Modeling Dynamic Systems

James D. Westervelt
Gordon L. Cohen *Editors*

Ecologist-Developed Spatially Explicit Dynamic Landscape Models

Springer

Modeling Dynamic Systems

Series Editors
Matthias Ruth
Bruce Hannon

For further volumes:
http://www.springer.com/series/3427

James D. Westervelt • Gordon L. Cohen
Editors

Ecologist-Developed Spatially Explicit Dynamic Landscape Models

Springer

Editors
James D. Westervelt
Construction Engineering Research Laboratory
US Army Engineer Research and Development Center
Champaign, IL, USA

Gordon L. Cohen
Information Technology Laboratory-IL
US Army Engineer Research and Development Center
Champaign, IL, USA

Operational copies of the models mentioned in this book are available at http://extras.springer.com/2012/978-1-4614-1256-4.

ISBN 978-1-4614-1256-4 e-ISBN 978-1-4614-1257-1
DOI 10.1007/978-1-4614-1257-1
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2012934178

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Editors' Foreword

We have given ourselves the job of helping to persuade you—a creative ecologist or social scientist—that you have all the necessary capabilities to begin capturing your unique expertise in simple, powerful simulation models that codify your knowledge into a computerized analytical tool. Your model gives you the opportunity to share your individual insights with your community of peers in the form of an easy-to-use, science-driven computer program that they can in turn examine, use, extend, and repurpose for their own work.

Simulation modeling is no longer the exclusive domain of elite computer scientists and programmers. Practical and expedient models now can be written without any mastery of low-level computer languages, numerical methods, or interface design. Simulation modeling platforms are now available that facilitate experimentation without bogging down the model builder in complicated software compiling tasks or graphical output issues. Powerful, user-friendly model-development tools have emerged—both open source programs and commercial packages—that can be mastered by anyone who has expert knowledge of a system, a fundamental understanding of desktop computers, and willingness to learn how to use software that is considerably less complicated than the everyday "office" applications that vex us all from time to time. You will find simulation modeling to be a gratifying and highly empowering skill if you are interested in:

- Harnessing computer power to reflect the implications of your intuitive understanding of a system, and make supportable predictions based on them.
- Verifying whether your understanding of a system can be codified in a way that replicates known system behaviors.
- Personally transcribing your intuitive expert knowledge into a transparent, science-based framework without asking computer programmers to intervene.

In the preface to this book, Dr. Bruce Hannon describes how he has encouraged a generation of social science and ecology students to climb the modest learning curve within a few class sessions, and then apply their skills to building operational simulation models in workgroups of two to eight. In his classes and the preface, Dr. Hannon emphasizes the benefits that students will gain by acquiring formal, but expedient, simulation modeling skills. He illustrates how an individual's deep understanding of a system's dynamics and behavior can readily be captured in a form that computers can process to unveil hidden implications of system processes that would otherwise probably evade conscious thought. Modeling enables you to do what you are good at—describing the system—while enlisting the computer to make supportable projections based on your expert knowledge.

This book is divided into two parts: (1) a technical orientation for prospective modelers and (2) examples of expedient operational models developed using the methods and tools described in Part I. The first part is intended especially for readers with no substantive experience in model building, but it includes insights that should benefit all modelers.

Chapter 1 addresses the topic of "modeling reluctance," for lack of a better term, that often inhibits ecologists and social scientists from acquiring model-building capabilities. As Dr. Hannon notes in his Preface, this inhibition can affect researchers like you, who have built a large store of technical expertise based on both direct observations from the field and an intuitive capability for drawing accurate inferences about future system behavior based on changes to the environment. If computer programming and higher mathematics are far removed from your daily practice, it is not surprising that you would be skeptical about how these disciplines might contribute to your work. Chapter 1 makes it clear that model building does not require high levels of computer or mathematical expertise and explains that modeling is already part of your everyday cognitive processes.

Chapter 2 describes a general process by which multidisciplinary groups may use relatively simple software tools to model relatively complex domains. It provides a general project roadmap to help multiple researchers from different disciplines work efficiently and harmoniously toward creating a rich simulation model in a very reasonable amount of time. These working guidelines have been used successfully at the University of Illinois for more than a decade to teach nonprogrammers how to develop dynamic simulation models working in a computer lab environment for several hours a week over a single semester. Most of the models presented in the second half of the book were created as class projects by multidisciplinary groups ranging from two to eight in size. Many of the team members were new to computer-based modeling.

Chapter 3 introduces you to NetLogo (Wilensky 1999), the model-development environment that was used to construct the models documented in the second part of the book. NetLogo is a free, public domain model-building software platform that enables you to describe the behavior of individuals within the spatial environment they inhabit. The individuals can interact with each other and their environment, and the environment itself may change according to its own dynamics. The chapter also provides grist for traditional computer programmers: a short introduction to Repast Simphony, a free, open source agent-based modeling package developed by Argonne National Laboratory, U.S. Department of Energy (http://repast.sourceforge.net/). Repast offers a migration path from simple NetLogo models to more challenging simulation modeling environments preferred by computer scientists. The fall 2010 release of Repast includes the ReLogo framework, which converts NetLogo models to Repast compatibility. Once converted, a computer programmer can then integrate the NetLogo model with other models, run the model on more powerful machines, and visualize and analyze model outputs in many useful ways not natively available in NetLogo.

Part II presents 11 simulation models, as documented using the Overview, Design concepts, and Details (ODD) protocol (Grimm et al. 2006). This protocol, developed cooperatively by 28 professional modelers, is a standardized model documentation specification intended to help a model builder clearly communicate the essential contents of a simulation model as well as its assumptions and scope. The purpose of the ODD protocol is to make the contents of a simulation model transparent to a reader who has some knowledge of the specific technical domain for which the model is built. This content format also makes it easy for the reader to evaluate similar models side by side.

All 11 models documented in Part II (Chaps. 4–14), and their data sets, are available for download and use.[1] These models were developed in NetLogo (Wilensky 1999), and your learning experience will be greatly enhanced if you load and run each model on your computer as you are reading about it in the book. You may download a full, operable version of NetLogo from http://ccl.northwestern.edu/netlogo/. All models presented in this book have been tested to run in NetLogo 5.0. Because NetLogo is programmed in Java (Oracle, Redwood Shores, CA), it operates on computers running Microsoft Windows, Macintosh OS X, or Linux.

Most of the models presented in Part II were developed and authored by students who took a University of Illinois spatial simulation-modeling course taught by Dr. Hannon, Dr. Charles Ehlschlaeger, and Dr. Jim Westervelt. They are grouped as individual-based models (IBMs) representing animal populations in the wild (Chaps. 4–8), a river nutrient model (Chap. 9), patch and inter-patch valuation models (Chaps. 10–12), and social models (Chaps. 13 and 14).

The first two models, fire ants (Chap. 4) and newts (Chap. 5), were developed by students in the class to explore, respectively, control measures for red imported fire ants (RIFA) in Texas and forecasting responses of striped newts to rainfall patterns in Georgia. In both cases, a pair of students new to simulation modeling turned literature reviews and interviews with experts into conceptual and then working models. The next two chapters consider the gopher tortoise, a species at risk, in the southeast United States. Chapter 6 captures a research effort that did not include direct involvement by an ecologist familiar with the gopher tortoise, but did involve experienced modelers. Conversely, the model in Chap. 7 was created by a team of ecologists familiar with the tortoise but without any experience in simulation modeling. This team quickly achieved proficiency with NetLogo.

The feral hog model described in Chap. 8 was developed by a team of seven graduate students, none of whom had previously written software. Their purpose was to test the hypothesis that adding a contraceptive program to an existing hunting policy would improve the control of wild swine on a military installation in Georgia. At one point during the course, the sound of virtual gunshots cracked out through the lab from computer speakers—NetLogo-generated hunters applying

[1] Operational copies of the models are available through http://extras.springer.com.

"control measures" during a demonstration of work in progress. The model proved so useful for testing the advantages of a proposed contraceptive program that one student, now the lead author of Chap. 8, was funded to further develop the model.

Chapter 9 explores nutrient cycling in the Mississippi River, taking into account the movement of nutrients via water currents in a pool on the river. River nutrients cycle through several trophic layers as the water flow moves components of the system downstream. This effort began with a nonspatial model written a decade earlier that was adapted to produce spatial output in NetLogo.

The next three chapters explore the value of land in terms of its contribution to the viability of a population. Habitat patches are analyzed in Chaps. 10 and 11. The first of those traces the lineage of populations in patches over time with respect to the original home of the original ancestors to reveal the relative connectivity among all pairs of patches. The second documents a model developed to reveal the relative value of each patch supporting a metapopulation in terms of sustaining the viability of the metapopulation. The intent of this second model is to support the development of an equation into which certain characteristics of patches, easily measured in the field, could be used to compute a "patch valuation" estimate. Chapter 12 looks at the value of land between patches for supporting inter-patch migration, which is necessary to connect populations into a broader metapopulation. This project translated a published supercomputer-based model into the NetLogo modeling system. The result is a very accessible model useful for experimentation and potential extension.

The final two chapters explore social science models that extend beyond natural ecosystems. Chapter 13 considers a model that forecasts urban residential growth patterns within a county based on the relative attractiveness of land to that growth. Domestic violence is the subject of the model documented in Chap. 14. The help-seeking behavior of violence victims is explored in a way that makes it possible to test policy impacts on violence rates.

Each of these models demonstrates how students and researchers have captured their understanding of dynamic spatial systems using a simulation modeling software package. The models make it possible for users to experimentally manipulate the system to test its response when subjected to alternate assumptions, conditions, or scenarios. Our hope is that these examples will encourage you to do the same!

Champaign, IL, USA James D. Westervelt
Gordon L. Cohen

References

Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, Giske J, Goss-Custard J, Grand T, Heinz SK, Huse G, Huth A, Jepsen JU, Jørgensen C, Mooij WM, Müller B, Pe'er G, Piou C, Railsback SF, Robbins AM, Robbins MM, Rossmanith E, Rüger N, Strand E, Souissi S, Stillman RA, Vabø R, Visser U, DeAngelis DL (2006) A standard protocol for describing individual-based and agent-based models. Ecol Model 198(1–2):115–126

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

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/

Contents

Contributors

Katherine R. Amato Program in Ecology, Evolution, and Conservation Biology, University of Illinois, Urbana, IL, USA
amato1@illinois.edu

Kimberly M. Andrews Savannah River Ecology Lab, University of Georgia, Aiken, SC, USA
Andrews@srel.edu

J.P. Aurambout Australian Department of Primary Industries, DPI Parkville Centre, Carlton, Australia
jeanphilippe.aurambout@dpi.vic.gov.au

Harold E. Balbach Construction Engineering Research Laboratory, U.S. Army Engineer Research and Development Center, Champaign, IL, USA
Hal.E.Balbach@usace.army.mil

Todd BenDor Department of City and Regional Planning, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
bendor@unc.edu

Jennifer L. Burton Department of Natural Resources and Environmental Sciences, University of Illinois, Urbana, IL, USA
jlburton@illinois.edu

Larry Carlile Fort Stewart Army Installation, Fish and Wildlife Branch, Fort Stewart, GA, USA

Gordon L. Cohen Information Technology Laboratory-IL, US Army Engineer Research and Development Center, Champaign, IL, USA
gordon.l.cohen@usace.army.mil

John Drake School of Integrative Biology, University of Illinois, Urbana, IL, USA
jedrake@bu.edu

Marina Drigo Department of Urban and Regional Planning, University of Illinois, Champaign, IL, USA
marina.v.drigo@gmail.com

Charles R. Ehlschlaeger U.S. Army Engineer Research and Development Center, Construction Engineering Research Laboratory, Champaign, IL, USA
Charles.R.Ehlschlaeger@usace.army.mil

Bruce Hannon Department of Geography, University of Illinois, Urbana, IL, USA
bhannon@illinois.edu

William W. Hargrove USDA Forest Service Southern Research Station, Eastern Environmental Threat Assessment Center, Asheville, NC, USA
whargrove@fs.fed.us; hnw@geobabble.org

Jeffrey Houser Upper Midwest Environmental Sciences Center, U.S. Geological Survey, La Crosse, WI, USA
jhouser@usgs.gov

Brian Ickes Upper Midwest Environmental Sciences Center, U.S. Geological Survey, La Crosse, WI, USA
bickes@usgs.gov

Richard F. Lance Environmental Laboratory, U.S. Army Engineer Research and Development Center, Vicksburg, MS, USA
Richard.F.Lance@usace.army.mil

Kevin Landwehr U.S. Army Corps of Engineers, Rock Island, IL, USA
Kevin.J.Landwehr@usace.army.mil

Paul L. Leberg Department of Biology, University of Louisiana, Lafayette, LA, USA
leberg@louisiana.edu

Ying Li Department of Crop Sciences, University of Illinois, Urbana, IL, USA
yingli3@illinois.edu

Bruce MacAllister Construction Engineering Research Laboratory, U.S. Army Engineer Research and Development Center, Champaign, IL, USA
Bruce.A.MacAllister@usace.army.mil

John Macey Fort Stewart Army Installation, Fish and Wildlife Branch, Fort Stewart, GA, USA

Benjamin Martin Department of Natural Resources and Environmental Sciences, University of Illinois, Urbana, IL, USA

William Meyer U.S. Army Engineer Research and Development Center, Construction Engineering Research Laboratory, Champaign, IL, USA
William.D.Meyer@usace.army.mil

Ariane Peralta Program in Ecology, Evolution, and Conservation Biology, University of Illinois, Urbana, IL, USA
alperalt@illinois.edu

Jon Petersen U.S. Army Corps of Engineers, Rock Island, IL, USA
Jonathan.W.Peterson@usace.army.mil

Tim Peterson Department of Kinesiology, Kinesiology and Community Health, University of Illinois, Urbana, IL, USA
tim.taiji@gmail.com

Aloah Pope Department of Natural Resources and Environmental Sciences, University of Illinois, Urbana, IL, USA
pope7@illinois.edu

Ewan Robinson Department of Geography, University of Illinois, Urbana, IL, USA
ewan.robinson@gmail.com

Bart Rossmann Applied Technology for Learning in the Arts and Sciences, University of Illinois, Urbana, IL, USA
bartmann@illinois.edu

Johanna Salzer Department of Pathobiology, University of Illinois, Urbana, IL, USA

Richard Sparks Illinois Natural History Survey, University of Illinois, Elsah, IL, USA
rsparks@illinois.edu

David Stigberg U.S. Army Engineer Research and Development Center, Construction Engineering Research Laboratory, Champaign, IL, USA
David.K.Stigberg@usace.army.mil

Elizabeth L. Sweet Department of Geography and Urban Studies, Temple University, Philadelphia, PA, USA
bsweet@temple.edu

Charles Theiling U.S. Army Corps of Engineers, Rock Island, IL, USA
Charles.H.Theiling@usace.army.mil

Tracey D. Tuberville Savannah River Ecology Lab, University of Georgia, Aiken, SC, USA
tubervil@uga.edu

Kranthi Varala Department of Crop Sciences, University of Illinois, Urbana, IL, USA
kvarala2@illinois.edu

James D. Westervelt Construction Engineering Research Laboratory, US Army Engineer Research and Development Center, Champaign, IL, USA
james.d.westervelt@usace.army.mil

Sheng Ye Department of Geography, University of Illinois, Urbana, IL, USA
sheng.y.ye@gmail.com

Yao Yin Upper Midwest Environmental Sciences Center, U.S. Geological Survey, La Crosse, WI, USA
yyin@usgs.gov

Chapter 1
Never Fear: You Already Model!

James D. Westervelt and Gordon L. Cohen

The task of writing simulation models to support environmental management decision-making has historically been assigned to software development specialists working in cooperation with a technical subject matter expert. The objective of model development was, and is, to capture the expert's knowledge of a system to provide a more formal description and analysis of the system. Because few ecological management professionals have the computer science training to direct the actual model-building effort, computer specialists have carried out most of that work. Common programming languages have included FORTRAN, C, Java, and Perl. Models based on statistical analysis have been developed using scripting languages with software packages such as R, SPSS, and SAS. Spatially explicit models incorporate geographical information systems (GIS) that provide scripting languages for executing map analysis. For purely mathematical models, there are programming tools included in industry-standard packages such as Mathematica and Matlab. Because of the need to recruit computer specialists for the bulk of model-building work on behalf of the subject matter expert, model development efforts have historically been costly and time-consuming. Ecological models considered to be the most useful over time have been given names, then reused, and maintained over many years. Not surprisingly, these successes tend to produce somewhat generic results.

Ecologists who have no formal training in model development—that is, most of them—have tended to consider the computer-based simulation model as a costly "black box" tool whose utility is marred by uncertainty about how it works inside. Consequently, most professional ecologists have chosen to ignore formal modeling

J.D. Westervelt (✉)
Construction Engineering Research Laboratory, US Army Engineer Research and Development Center, Champaign, IL, USA
e-mail: james.d.westervelt@usace.army.mil

G.L. Cohen
Information Technology Laboratory-IL, US Army Engineer Research and Development Center, Champaign, IL, USA
e-mail: gordon.l.cohen@usace.army.mil

J.D. Westervelt and G.L. Cohen (eds.), *Ecologist-Developed Spatially Explicit Dynamic Landscape Models*, Modeling Dynamic Systems,
DOI 10.1007/978-1-4614-1257-1_1, © Springer Science+Business Media, LLC 2012

of their management domains and make decisions based on their own scientific expertise, experience, and intuition. We might say that ecologists are typically most comfortable using the creative part of their brain, their right brain. Thinking with this part of the brain opens the conscious to the unregulated wanderings and musings of the subconscious where experiences are folded together to provide us with deep understandings of systems. This results in our ability to intuitively forecast consequences of actions on or changes to the systems with which we are familiar. Through this type of intelligence, instinct, and good luck, most ecologists can succeed, excel, and greatly prosper at their work. For a scientist, however, using only this approach has its shortcomings. One significant deficiency is that the individual's professional knowledge itself emerges from a kind of black box—a complex but subjective thought process in which the system is captured only informally, as far as the rest of the world knows, possibly using unquantifiable assumptions or undocumented intuitions. This is where left brain thinking, involving logic, classification, scheduling, process, and procedure, becomes valuable. By carefully looking at and describing the parts of a system, it becomes possible to build a more comprehensive understanding of the cause-and-effect relationships underlying the behavior of the whole system. When we are able to capture the assumptions and dynamics of a system formally and clearly, they become available for review by anyone who can read the language in which the knowledge is encoded. That encoding has traditionally been accomplished in print, using the language of the discipline. The same information also can be captured in computer languages, and this is highly useful because computers are very good at executing their instructions to reflect the behavior of a captured system when the state of the model is altered. When used together, captured right- and left-brain thinking can provide much more complete and compelling insights than using either exclusively.

Although the practice of modeling may seem abstruse to the reader who is not a computer programmer, we will venture to point out that you are already an expert modeler! Humans, as well as all higher animals, must reliably model the world around them in order to survive. Grazers, for example, must retain a mental image of the seasonally changing locations of food, shelter, and water. Predators and prey must develop cognitive models of the "battleground" where hunting and foraging take place. Each must be able to sense the probability of success or failure in satisfying the fundamental requirements for survival and procreation. The deer, on one hand, has a model of how close it can allow the wolf to approach in the current terrain and still be able to flee to safety; on the other hand, the wolf has a model of how to use the same terrain, wind direction, and other factors to approach the deer with stealth in order to overtake it. A tourist visiting in New York City must develop an "on the fly" model for how to efficiently cross a busy street without being run over by a yellow taxi.

Consider the components that comprise a "geospatially explicit model" of a baseball outfielder as a towering fly ball soars toward the wall in left-center. The player must rapidly calculate trajectory and peak altitude in order to determine whether to field the ball on the fly or a bounce, knowing both the static location and height of the wall and also the dynamic location of another outfielder who may be running on a converging path. Temperature affects air density and, thus, the drag of friction; and the cloud cover may or may not provide additional visual information. All these data streams feed an ad hoc model that the fielder uses to coordinate running across a landscape and reaching into space in the hope of putting out a batter, and perhaps instantly hurling the ball to an infielder to keep a runner from advancing.

Or consider a more impressive modeling project: a baby learning to walk. Lacking even the basic tools represented by a significant spoken vocabulary, the infant builds rudimentary models of his or her body in space and time, refined and combined over a year on the planet, to fully repurpose a body that was seemingly designed for a prone, static existence. This new mind, unimpeded by ideas of what may not be possible, exercises a growing spatial awareness and flaccid neck muscles to balance an outsized head upright. Within a few months, the baby can balance the whole upper body in a seated position—no hands! And then, through imitation, trial, error, continuous observation, and revision and combination of "sub-models," uses those free hands and higher-resolution spatial models to grasp objects and parental appendages in the environment. Legs learn to perambulate in response to forward guidance by a parent; lower body strength grows; sub-models of vertical balance are revised as the baby learns how to stand stably with a higher center of gravity than just a few weeks before. Somehow, using observational and experimental methods that cannot be adequately communicated in chapters of technical writing, the infant applies spatial and nonverbal conceptual models to learn the exquisitely complex task of combining bodily motion, balance, and controlled falling into standing and walking at will. The physics and calculus required to model this task for a robot are highly challenging to this day, even after decades of engineering research dedicated to that purpose.

Although the human ability to create, refine, and apply conceptual models is formidable, and has helped us to survive and prosper through the ages, it has two significant limitations: our models are very difficult to accurately communicate, and we are not good at predicting the behavior of a system when it includes a feedback loop. Both of these limitations can be powerfully addressed using mathematics, formal logic, and computer software.

The limits of fully communicating our conceptual models often become apparent when we initiate the *because-I-said-so* "sub-model," or someone uses it with us. Closest to home for a parent are the continual cases in which attempts to communicate our models of, say, good nutrition or personal hygiene are reflexively challenged with the question, "Why?" In such cases, *because I said so* may suffice as a functional sub-model for practical purposes. This same "sub-model" also is invoked by many professional practitioners, albeit using more customer-friendly phrasing, when a conceptual model is too nuanced to communicate. The physician, the attorney, and the financial advisor all rely on models of their technical domains, developed through formal education and experience, to effectively advise their clients. They pronounce a diagnosis, or declare a legal strategy, or propose an investment plan based on a tacit agreement that, for the most part, these are valid recommendations *because I said so*.

Most conceptual models, whether professional or personal, are not developed using mathematics and are not shared with others as precise, formal descriptions. These models are created primarily through training of the neural network hosted within our brains to discover and retain associations of information (e.g., a patient's symptom) with an explanation or response pertaining to it (e.g., a diagnosis and treatment). However, while we can help someone to understand a model through observation, trial and error, and repetition, it is very difficult to explicitly and fully communicate a model, even for something as seemingly simple as distinguishing a cat from a dog. It is simply more practical to repeat the training course for each of our children, complete with illustrations and repetitions, than it is to prepare a definitive documented model of comparative animal morphology and behavior.

Unfortunately, that practical approach is not sufficiently rigorous for the effective documentation and transfer of knowledge according to the scientific method. It is likely that few center fielders could formally document the process of trying to snag a 385-ft blast heading toward the fence, and most people probably cannot clearly explain their conceptual model for distinguishing a dog from a cat. The physician or attorney may have more success describing the model for a recommended treatment or a legal strategy since their professions depend to a great extent on systems of logic, but few could provide a succinct model for applying their professional judgment given all the changing assumptions and variables each case imposes. Inevitably, people with expert knowledge are faced with a gap between what they know "in their bones" and their ability to convince their clients or colleagues of its validity.

Fortunately, we have formal logic, and derivative mathematical and symbolic forms of it, which can help guide us from what is known toward new information as yet unknown. Formal logic provides uniform standards for reasoning and critical thinking, and accepted methods for applying logic are invaluable for documenting the validity of thought processes and identifying fallacies in them. Furthermore, powerful quantitative tools such as statistics, matrix algebra, and calculus extend formal logic into highly abstract realms of mathematics and science that are otherwise impossible to penetrate. These tools and their underlying systems of logic have made it possible to capture professional expertise as highly explanatory models of physical systems. Using various computer programming languages, expert knowledge can be encoded to create automated tools that simulate the consequences of altering the system's assumptions, parameters, and variables. These *simulation models* are also very good at processing the effects of feedback loops in systems—something of which the human mind is much less capable because the results may appear too complex for comprehension. An additional benefit of simulation models is that they produce repeatable results.

As indicated at the beginning of this text, however, modeling has generally remained a "sandbox" for computer programmers and other researchers who have the skills to translate conceptual models into mathematical algorithms, and then into computer programming languages. Practitioners who possess both subject matter expertise and excellent model-building skills have been almost as rare as alchemists; the communication gulf between scientist and modeling team has been almost as wide as the one between, say, attorney and client. The necessary division of labor between the scientist and the modeling team's mathematicians, statisticians, and programmers has encumbered both model development and adoption by interested practitioners. Why is this so?

First, creating models has typically been time-consuming and costly because of inefficiencies inherent in translating one expert's intuitive knowledge of system dynamics into the precise language of algorithms, and then translating the algorithms into computer code. Each translation has historically been executed by a different expert. In order to ensure that nothing has been "lost in translation," several iterations of revision and verification may be necessary before all contributors are confident of the model's scientific content and operation. The process competes with many other research activities for sufficient funding, personnel, and time allotted to produce results.

Second, even after the model has been verified and formalized by the development team, there is still the matter of validation (i.e., proof of accuracy). Without validation, field ecology peers who cannot readily look inside to evaluate it for themselves often regard the model as a "black box." On one hand, the prospective user deserves to know whether the model is accurate; on the other hand, the subject matter expert is put in a position of being considered guilty until proven innocent by second or third opinions—not unheard of in the professional world, but quite far from standard practice, too.

An unfortunate implication of this dilemma is that the highly technical aspects of model development have contributed to the alienation, or perhaps intimidation, of practitioners in the field who might greatly benefit from using explanatory simulation models. However, technology is on the side of scientists who have an interest in simulation modeling but no practical way to use it. The growing availability of open source software tools, and methods for using them to capture expert knowledge of system dynamics, now make it possible to develop spatially explicit simulation models without formal training or programming skills.

This book is written for ecologists and students of ecology who are interested in the idea of capturing and sharing their own undocumented conceptual models of natural systems using simplified software tools and proven collaborative methods of development. The immediate benefit of creating this type of model is that internal expert knowledge becomes clarified and quantified as a decision-support tool the scientific content of which may be reviewed by others with related expertise. These models may be revised and extended with relative ease, and some may even be adapted or repurposed beyond their original intent without starting again from a blank slate. The first part of the book summarizes current state of the science and art of ecological simulation modeling. It includes chapters dedicated to a survey of landscape modeling environments for users with no formal programming experience and methods for managing a multidisciplinary ecological modeling project. The second part of the book documents 11 case studies where expedient ecological simulation models have been developed and applied by university graduate students. These applications are used to support management activities ranging from species at risk and nutrient flows in rivers to food distribution and social services.

The editors of this collection have two objectives for the book. The first is to outline an expedient and effective methodology for fielding useful ecological simulation models. The second is to inspire readers with the confidence that it is within their grasp to create and use computer-driven tools that help to clarify and extract their professional expertise, and share it with their community of peers. We sincerely hope that this guide contributes to the advance of effective environmental management practice for the purpose of improving ecological sustainability.

Reference

Wilensky U (1999) NetLogo. Computer software. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, Jan 2011. http://ccl.northwestern.edu/netlogo/. Accessed 01/2011

Chapter 2
A Collaborative Process for Multidisciplinary Group Modeling Projects

James D. Westervelt and Bruce Hannon

2.1 Introduction

Although a virtually unlimited number of ecosystem management issues may be illuminated using small, expedient simulation models developed by one person or a few, there are many cases where much more complex problems can be efficiently modeled by a relatively larger team that spans several disciplines or technical domains. This chapter describes a process for conceiving, coordinating, and launching such a model development initiative using the same simple software platforms described elsewhere in this book.

We have successfully taught ecosystem modeling to groups of university students as a three-stage process, with a sequence of steps comprising each stage. By adhering to this process, our students have developed highly utilitarian ecological simulation models that are based on real-world data and specialized technical expertise. Many of the models documented in this book were developed by students. Our model-development process can be outlined as follows:

1. Prepare to model
   (a) Identify objectives and scope
   (b) Identify available resources (personnel, expertise, time, software, hardware)
   (c) Consider group dynamics (including ownership issues)

J.D. Westervelt (✉)
Construction Engineering Research Laboratory, US Army Engineer Research and Development Center, Champaign, IL, USA
e-mail: james.d.westervelt@usace.army.mil

B. Hannon
Department of Geography, University of Illinois,
220 Davenport Hall, 607 S Mathews, M/C 150, Urbana, IL 61801, USA
e-mail: bhannon@illinois.edu

J.D. Westervelt and G.L. Cohen (eds.), *Ecologist-Developed Spatially Explicit Dynamic Landscape Models*, Modeling Dynamic Systems,
DOI 10.1007/978-1-4614-1257-1_2, © Springer Science+Business Media, LLC 2012