The Canadian Society for Bioengineering
The Canadian society for engineering in agricultural, food, environmental, and biological systems.

La Société Canadienne de Génie Agroalimentaire et de Bioingénierie
La société canadienne de génie agroalimentaire, de la bioingénierie et de l'environnement

CSBE | SCGAB

Paper No. 06-302

Object-Based Modeling and Simulation: What, Why, and How

Tania R. Lanphere and Robert Kok

Bioresource Engineering, Room MS1-027, 21111 Lakeshore Road, Ste. Anne de Bellevue, Quebec H9X 3V9

Written for presentation at the
CSBE/SCGAB 2006 Annual Conference
Edmonton Alberta
July 16 - 19, 2006

Abstract

In an object-based model each component is represented as a discrete entity. The spatial resolution of the model can be set at any desired level and thus describe a given system in many different ways. The sophistication of the model can be determined by adjusting number of attributes used to describe each object. Relationships between objects are described with rule based expressions, and the combined activities of all the objects give rise to the global dynamics of the system. To elicit these dynamic, the interaction of the objects is typically “acted out” in a simulation for a number of equal time intervals under the influence of forcing functions and disturbances (e.g. solar radiation, temperature, rainfall, earthquakes). Object-based modeling has both advantages and disadvantages, and thus is not applicable for all modeling / simulation projects. However, it is very useful for modeling complex systems, both existent and hypothetical, and for investigating general, complex system problems and methods using *virtual systems*.

Keywords: ecological modeling; object-based modeling

Introduction

One problem with many ecosystem related research projects is the amount of time that field and bench scale experiments can take. Studying any process that takes more than a few years requires a great deal of organization and often a large number of people. It is also difficult to perform studies that apply to more than one type of ecosystem. Thus, computer models are very useful tools for studying ecosystems, their dynamics and processes, and predicting ecosystem response to certain situations. These models can be used to represent existing physical ecosystems, hypothetical physical ecosystems, or virtual ecosystems. They may be intended to study one particular aspect of a system, the entire system, or ecosystem dynamics and processes in general. Accordingly there are many different models and many different modeling approaches. One such approach is object-based modeling.

Object-based modeling

In an object-based model each system component, biotic and abiotic, is represented as a discrete entity, or object. The modeler can set the spatial resolution of the system representation at any desired level and thus describe any given system in many different ways. For instance, an ecosystem might be modeled at the micron scale, the millimeter scale, the meter scale, etc. Analogously, the objects in the model would correspond to components the size of bacteria, ants, wolves, etc. The modeler also determines the sophistication of the model by setting the number of attributes attached to each object. One might only pay attention to the energy flow through a system, for instance, and thus give each object only two attributes: individual age and total energy content. In this approach to modeling, the relationships between objects are described with rule-based expressions which can, for example, relate to food preferences and feeding behavior, mating preferences, health status, nesting patterns, etc. For abiotic components they can relate to fluid flow, diffusion, chemical oxidation, etc. Of course, inter-object relationships specified in the model must correspond to the attributes of the objects that are defined. Typically, the interactions of the components are “acted out” in a simulation for a number of equal time intervals (the model is “implemented” in simulation) under the influence of a number of forcing functions and disturbances (solar radiation, temperature, rainfall, earthquakes). The combined activities of all the components then give rise to global dynamics. Of course, nothing can be “seen” in the global dynamics at frequencies higher than the Nyquist frequency corresponding to the time interval and, equivalently, nothing can “seen” in the system at a scale smaller than that corresponding to the resolution. As well, only phenomena that are related to the attributes can be observed. Of course, these limitations are common to all modeling approaches.

One great advantage of object-based modeling is its close correspondence to human perception of the world. We naturally perceive the world as a complex, interacting set of objects which we describe in terms of a number of attributes like size, shape, and color, with each of these having a value. Thus, the human perceptive system naturally identifies eyes, ears, arms, legs, heads, gender, etc. on components which it identifies in its environment, and assigns values to these (e.g., my mother’s eyes are *blue*). All human biological taxonomies and other classification schemes are structured like this and it seems to be human nature to operate in this manner. (It should be noted that this assertion is in no way related to any assertion about the “true nature” of that environment.) Hence, object-based modeling is seen to be less abstract than many of the other modeling approaches, such as the use of differential equations to describe the flow of heat and fluids. This is directly related to its second great advantage: to use this method there is no need to have knowledge/understanding of the larger system. With other words, it is sufficient to have knowledge at the micro scale to obtain results at the macro scale; if one understands the individual chicken, there is no need to have knowledge of chickens at the flock level. Flock behavior will emerge naturally from the myriad interactions between chickens; watershed dynamics will emerge naturally from knowledge of the topography and soil, combined with the rule “water flows downhill”. At the same time, when one does have macro-scale knowledge, then seeing overall system dynamics emerge from an object-based model that closely mimics known behavior certainly helps to inspires confidence in the model’s validity!

Evidently, object-based modeling does not only have advantages. One of its main disadvantages is that it is computationally heavy. This implies that the computer hardware required for simulations needs to be large and fast. A “small” ecosystem simulation typically requires a machine with multiple gigabytes of memory, and can easily take days or weeks (SunFire 880 server running Solaris). Large simulations must be run on supercomputers. The method has this resource-intensity in common with modeling methods such as finite elements and finite differences. (It could, in fact, be argued that the object-based approach is merely a refinement of finite elements, with many different types of multiple-attribute elements being defined in the system, and the possibility of elements being mobile. Indeed, the boundary between finite elements and object-based modeling is not really that well defined.) Of course, as progress is being made in machine development, computational requirements will become easier to meet. Hence, this problem may become insignificant in another twenty years. A second disadvantage of the object-based approach can be its very character! Thus, it is frequently actually preferable to deal with a system in terms of a greater level of abstraction. In such cases an object-based approach would merely complicate the issue without yielding a compensating advantage. For example, to model a homogeneous beam of simple geometry which is being subjected to a sinusoidal forcing function, it is probably preferable to use a differential equation approach since the solution to the problem is well known and readily available, and easily and quickly yields all the results that are typically desired.

Under which circumstances, then, is the object-based modeling approach most applicable? The simple answer to this is: "when the situation is more complicated than is optimal for the other methods". With other words, when a system is composed of a number of different types of components which are also described differently (different attributes and a different number of attributes), when the components are spatially mixed in or on a space of more than two dimensions (e.g., where a landscape might have a dimension of 2.4), when the components are mobile in or on a space (e.g., cells traveling through a network of blood vessels), when there are extensive interactions between the different types of components or between components of the same type – such interactions may be direct (e.g., feeding) or indirect (e.g., via pheromones), when populations of components might increase or decrease due to "birth" and "death", and when components might undergo adaptation and/or mutation and evolution. Systems that meet any or many of the above criteria can often be classified as "complex systems" and, depending on whether they are self-maintaining and/or self-reproducing, they might also qualify as "biosystems". Hence, the object-based approach is generally most applicable when the system is rather complex. Suitable targets for modeling with this technique include ecosystems as well as smaller biosystems like cell cultures and individual organisms, as well as complex arrangements of purely abiotic components like watersheds and large irrigation systems. In terms of subject system class, object-based modeling is appropriate for dealing with existent systems (to develop, for example, control strategies for water allocation), for working with hypothetical systems (e.g., for design purposes), as well as for investigating more general, complex system problems and methods (with the use of purely *virtual systems* such as *virtual ecosystems* – these are not "models" as such since no particular system is being emulated – rather, they are systems in their own right).

Usually, an object-based model (or virtual system) is composed in three forms before it is implemented within a simulation framework: conceptual form, representational form, and computational form. The computational form can be written in any suitable computer language. The most obvious type of language might seem to be an object-oriented one like SmallTalk or C++ and much of our earlier work was based on these. However, in our experience, there are some problems with these, especially when the system is larger (e.g., more than a million objects). Thus, we have found the execution speed of programs in SmallTalk and C++ (compiled) to be quite low as compared to those written in Fortran. As well, we have found that programs written in C execute slower than those written in Fortran, for similar component population sizes. Thus, our approach has become to write object-based models and their accompanying simulation frameworks in Fortran. This results in code that is very transportable between different classes of machines (portables and desktops, workstations like the Dell 670 or Sun FireBlade, moderate-sized servers like the SunFire 880, and truly large computers). Thus, one can easily prototype, verify the code, and validate the model on an office machine with small population sizes, and then use an optimizing compiler for production runs on a larger machine. Once the methodology is developed, Fortran can be quite efficient at both storing and manipulating large, sparse matrices in which the elements are interconnected via a fairly intricate web of interactions which can be present in the form of rules, mathematical relationships, etc. It is surmised that even greater computational efficiency might be attained by writing some of the base routines in an assembler-level language, an approach utilized by the creators of commercial products such as Matlab. This is, however, quite costly and normally requires the involvement of professional programmers. As yet, we have not done this.

Example applications

We have used object-based modeling (and simulation) for a number of research projects in which we attempted to establish a scientific basis for the engineering design of ecosystems. In all these projects we have relied on the use of purely virtual ecosystems which were subjected to various forcing functions and disturbances resembling naturally-occurring ones to some degree. Thus, we typically define a number of component types (biotic ones corresponding to various plants, animals, and microbes, as well as abiotic ones corresponding to soil elements, ponds, streams, irrigation equipment, control elements, etc.) and set up interaction matrices between these. The type definitions can be as simple or as detailed as one wishes. We then initialize a system or multiple systems by creating populations of instances of each type, and set the system in motion. In one project the types were defined in considerable detail, with the “animals” displaying nesting behavior, feeding behavior, etc. and the plants dropping leaves or needles and distributing seeds (Lael Parrott, PhD). For system initialization the instances were created at different ages and with different compositions, i.e., some animals were “fat” and “old” when they were initially created, whereas others were “lean” and “juvenile”, etc. The whole system was subjected to seasonal variations in temperature and light intensity. For another project (Robert Molenaar, PhD), the types were much simpler but extensive, active control was added to the ecosystem to improve its stability, robustness, and longevity. Four levels of control were incorporated: basal, instinctive, Pavlovian, and cognitive. A project presently underway (Tania Lanphere, PhD) is also based on fairly simple components, with the objective of compiling enough of a data base on this kind of system to allow for the induction of rules for design and initialization. The intent is to shed light on how to define sets of component types that will, together, form a system of a specified stability and display a particular kind of dynamics, when initialized at a certain point in state space. In a related project, just finished, the biotic components were given the capacity to adapt (Marc Abbyad, MSc) to examine how that would affect species success. Although not directly based on modeling and simulation, the research carried out in this program by Grant Clark (PhD) was oriented to defining and describing the conceptual framework in which this research is being carried out.

Overall, the intent of the research is to establish a scientific base for the engineering of complex systems, specifically biosystems, and more specifically ecosystems. The approach being followed is to create object-based virtual ecosystems, to implement these in simulations, and then analyze the results.