

Incorporating Complexity in Ecosystem Modelling

Lael Parrott

*Dept. of Agricultural and Biosystems Engineering
McGill University
Quebec Canada*

Email: laelparrott@hotmail.com

Robert Kok

*Dept. of Agricultural and Biosystems Engineering
McGill University
Quebec Canada*

Email: kok@macdonald.mcgill.ca

Abstract

This article is a discussion of how the field of ecosystem modelling is being affected by the adoption of ideas arising from complex system studies. The modelling process is presented as being composed of four stages, starting with a modelled system, which is then depicted in turn by conceptual, representational, and computational models. The way that each of these stages is affected by conceptualizing an ecosystem as a complex system is discussed, with reference to current trends in ecosystem modelling. In particular, the recent emphasis on the use of “object-based” models, in which an ecosystem is represented at a high level of resolution as a collection of many thousands of interacting components, is presented as being an obvious method by which to reproduce complex dynamics in computer-based simulation. Three main types of object-based models are identified: individual-based models, agent-based models, and cellular automata. These three types are discussed with reference to examples from the literature. Several different computational approaches and programming platforms that are applicable to object-based ecosystem modelling are then reviewed. The paper’s target audience is anyone who desires, for teaching or research purposes, an overview of the various modelling methods (and their advantages and disadvantages) that are currently being used to represent ecosystems in the context of complexity research.

1. Introduction

In the 1950's, early cybernetics theorists postulated that it was fundamentally impossible to study a *complex system* by dividing it up into components and then analysing each part in isolation (see, for example, 1). A complex system, unlike a simple one, was viewed as being an entity whose global behaviour was somehow “more than the sum of the operations of its parts”. Today, a complex system is still most commonly defined as a network of many components whose aggregate behaviour is both due to, and gives rise to, multi-scale structural and dynamical patterns which are not inferable from a system description that spans only a narrow window of resolution (adapted from 52). Contemporary complex system studies, the origins of which can be traced back to early cybernetics research, involves the application of recent developments in fields such as non-linear physics and modern dynamical systems theory to the analysis of a broad range of natural and artificial systems. The field is, therefore, highly multidisciplinary, bringing together researchers in all specialties, ranging from economics and social policy to biology, physics and modern visual arts (19; 73; 65).

In general, most systems that are described as being “alive” or “intelligent” are considered to be complex, as are most loose assemblages of such systems. Genetic networks, biomes, minds, and human societies are all typical examples. The theoretical study of complex systems has been mostly focused on their organisation (defined as per 43: *the set of relations that determine the kinds of interactions and transformations within a system*), and on the arrangements that contribute both to the development and persistence of certain features within a given organisational envelope. In this context, it is the relationships between components (i.e., structure), rather than the components and their properties (i.e., composition) that are most significant. This emphasis on structure over composition is what makes the analytical approach of complex system studies so applicable across disciplines, since so many different types of systems can be characterised with similar analytical tools.

The analysis of ecosystems in this context allows an ecosystem to be viewed abstractly as a distributed network of sparsely connected components, many of which interact in ways that can only be described by highly non-linear relationships. Thus, as for any complex system, the aggregate behaviour of small scale components in an ecological network is seen to give rise to higher-level features and patterns, the emergence of which can not usually be predicted by a reductionist analysis. In this manner, the study of ecosystems is placed within a broader theoretical context, in which ecosystems can be subjected to the same methods of characterization, modelling, and description as is used for other complex systems, and the similarities (and differences) between ecosystems and other networks can be elucidated.

Due to the potentially fundamental impact that concepts from complex system studies may have on the way ecosystems are perceived, there are authors who have stressed the need for increased collaboration between ecologists and complexity theorists (34). To date, there are a number of key (and inter-related) features have been identified that seem to be common to all complex systems, and which are potentially relevant to the study of ecosystems. They include: temporal and spatial self-organisation (18), emergence (2), adaptivity (25), critical levels of connectivity (32), and, in living systems, autopoiesis (43). All of these can also be shown to be features of ecosystems. For example, the seemingly random, or erratic, behaviour of individual organisms, in conjunction with random environmental influences, can produce persistent, self-organised structures and dynamics at the population scale (e.g., dispersal or flocking patterns, quasi-equilibrium, resilience or persistence of species in ecological communities). These, in turn, affect the behaviour of the individuals in the populations, giving rise to emergent feedback loops and other autopoietic, or self-regenerating, structures and functions (8). The adoption of the idea that there may be critical levels of connectivity in

ecosystems is also changing the way that food webs and landscape patterns are analysed (15; 34). A significant impact of this has been the recognition that the inclusion of spatial elements in an ecosystem model can have a radical effect on the predicted dynamics, often moving a system from an equilibrium state in the non-spatial case, to chaos or some other complex regime (31;67). In the past, such dynamics may have been rejected as “invalid” or “unstable”, whereas now these patterns are believed to be closer to what likely occurs in physical reality.

2. Modelling an ecosystem as a complex system

The integration of the fields of complex systems studies and ecology is having a cascading effect throughout the entire ecosystem modelling process, beginning with an alteration of the commonly accepted conceptual model of an ecosystem, and resulting in the development and adoption of entirely new computational approaches. The modelling process (Figure 1) is presented here as being made up of several stages, beginning with a modelled system that is analysed according to a particular conceptual approach, and ending with a computational model that is implemented in simulation. As illustrated in Figure 1, when an ecosystem is analysed and subsequently modelled in the context of complex system studies, all of these stages are affected.

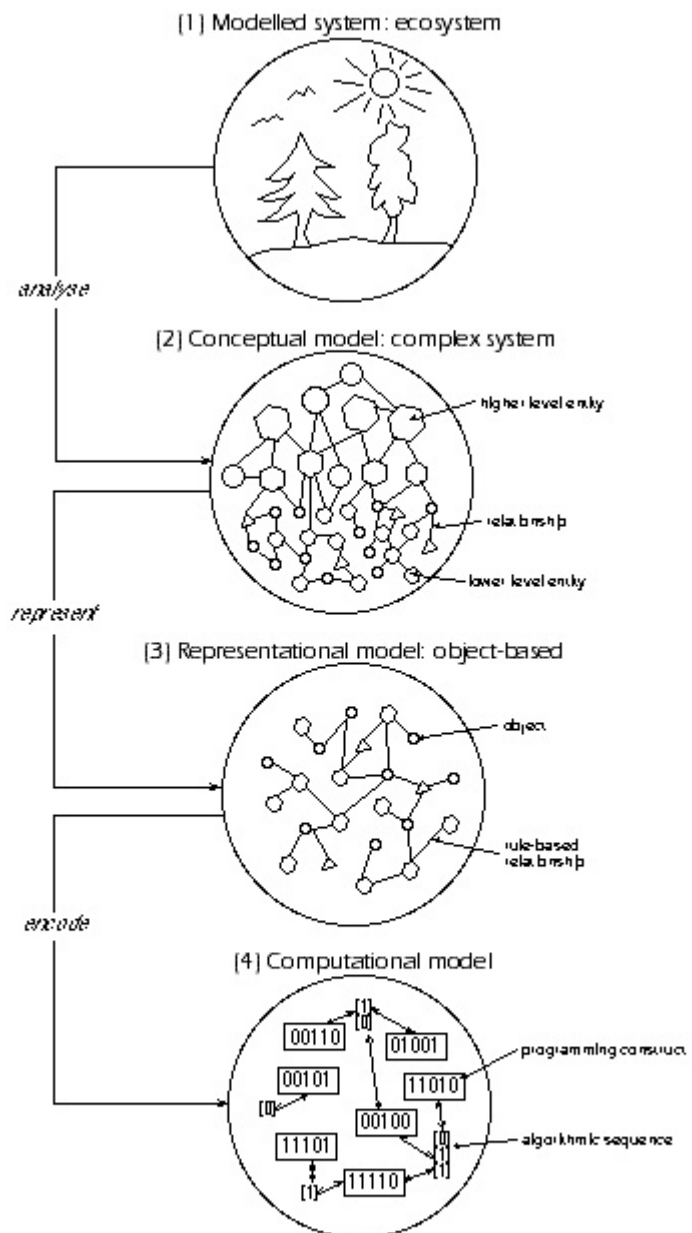


Figure 1. The modelling process

2.1 *The conceptual model*

In the context described above, an ecosystem is conceptualized as a complex system, i.e., the complex system is used as the conceptual model and all observations of the modelled system are interpreted in this sense. A key feature of this conceptual model is organisational hierarchy: every entity in a complex system can be described as being a conglomerate of smaller components (a country is a collection of provinces; a population is a collection of individuals; etc.), and the state of every higher level entity is the result of the states of the lower level components of which it is composed. Although living systems have always been described in terms of nested levels of assembly: cells << tissues << organs and other internal systems << individuals << populations << communities << ecosystems << biomes, the study of such systems has typically been concentrated on a fairly narrow (and therefore manageable) range of levels, which were treated as being distinct from one another. Thus, whereas in most conventional conceptual approaches, ecosystems are studied at only one hierarchical level, the approach of complex system studies provides a framework in which entities at different levels, and the relationships between these, can be accommodated. In this manner, the conceptual model which is used to describe ecosystems when they are analysed according to the approach of complex system studies engages a somewhat larger “depth of field” than that of conventional models. This seemingly simple modification has had a dramatic influence on the way in which ecological dynamics are explained, and, consequently, a large number of global phenomena have been reinterpreted as events that emerge from inter-hierarchical interactions. Examples include: the colonisation amongst groups in a species (2), the almost periodic boom-bust cycles in a predator-prey relationship, the flocking of birds (62), and the natural tendency of organisms in a population to be distributed log-normally over a terrain (71).

2.2 *The representational model*

Since a complex system is viewed as being a network of interacting components, the most obvious approach to use in representing it is one in which each component is modelled as a discrete entity. This approach is referred to as “object-based”; the resulting model is then referred to as an “object-based model”. In an object-based model, components in a system (as they are recognisable at a given degree of resolution, usually at one of the lower levels in the organisational hierarchy) are each depicted separately, and the relationships between them are described by rule-based expressions. Subsequently, when this type of model is encoded in computational form and then implemented in simulation, the combined activities of the objects are used to reflect the global dynamics of the system. Thus, in imitation of physical reality (according to our current understanding of the dynamics of a complex system), the progression of a system’s global level dynamics is not pre-specified as such, but instead is allowed to emerge as the result of events occurring at the object-level. In ecology, this approach is commonly referred to as “bottom-up” modelling (33).

An important aspect of object-based models is that the object’s internal transition functions, and relationships with other objects, are described by rules; and an object’s state is described by the values of a number of variables. Unlike models based on difference equations, in which continuous variables are discretized, in an object-based model, the variables treated by rule-based expressions are not necessarily continuous, and may take on just a few values. Additionally, changes in an object’s state at a discrete moment in time are determined by both these rules, and, often, the result of a chance process (e.g., the value of a randomly generated number). In this manner, the discrete mathematics of object-based models mimics the element of chance in the natural world that leads to the uncountable number of “frozen accidents” in history, from the level of quantum mechanics, to genetic mutation, to seemingly irrational social behaviour (12).

The term “object-based” is used here to refer to any type of model that meets the above description, regardless of what type of physical components the “objects” are supposed to represent. In models of ecosystems, the “objects” most often depict organisms or small spatial patches. This approach has become increasingly popular in ecosystem modelling, since it allows for the investigation of the lower level mechanisms that give rise to the development of higher level structural and dynamical features in ecosystems. For example, the social behaviour of ants has been represented with models in which a large collection of individual organisms move about under the influence of a field of morphogens (whose concentration is defined locally, i.e., at the object-level) while at the same time altering the global shape of that field as a result of their passage through it (47).

2.3 The computational model

In order to be used in simulation, an object-based representation must be encoded in computational form (Figure 1), in which the state of each object is stored in some form of “programming construct”, and the rules with which object interactions are specified are written as algorithmic sequences. The resulting computational model is then implemented in simulation in order to reproduce the dynamics of the modelled system. The choice of encoding method (i.e., programming techniques, etc.) is not necessarily determined by the type of representational model used, and is very dependent upon both the skills and objectives of the programmer. As discussed in Section 4, many object-based models are encoded with object-oriented programming languages, which have a number of inherent advantages over other languages with respect to discrete entity modelling.

3. Types of object-based ecosystem models

There are many different types of object-based models that have been developed to represent ecosystems, and these differ with respect to the kinds of components represented by objects, as well as the choice of rule-sets. Many of these types of models were originally developed by researchers in the field of artificial life, a branch of complex system studies oriented toward the study and creation of new life forms. For the purposes of this review, the various types of object-based representational models used in ecology have been divided into three categories: individual-based models, agent-based models, and cellular automata. In individual-based ecosystem models, objects represent organisms; in agent-based models, objects represent life-like “agents” that have the capacity to evolve or adapt their behaviour; in cellular automata, objects are arranged on a lattice, and usually represent small spatial elements in an ecosystem. Examples of these three types of object-based models are given below, followed by an overview of some of the simulation platforms that have been developed to facilitate the encoding of object-based representational models in computational form. This review is by no means exhaustive; for each type of model, only a few of the most illustrative cases from the literature are used to highlight the basic concepts.

3.1 Individual-based models

With an individual-based model, an ecosystem is represented as a large collection of interacting organisms. Population-level dynamics, therefore, are elicited in simulation by summing the collective activities of many individuals. Thus, with this approach, the basic unit for modelling is the organism, which is portrayed as a discrete object, whose state is usually described by a number of attributes. An organism’s behaviour is modelled with various rules that represent its potential interactions with other individuals and with its environment.

Individual-based modelling has increased in popularity in recent years, partly due to the greater computational power available to researchers (which makes this approach more feasible than it was in the past) and partly due to an increasing recognition of the advantages of this approach in comparison with population-based models (30; see also: 17, for an overview of the growth of individual-based modelling in ecology). There are, for example, a number of phenomena such as the development of subgroups within a population, and the interactions between these and subgroups of other species, that occur at a degree of resolution which is finer than that of a typical population-based model and that are therefore much easier to reproduce with an individual-based model. Other examples of such phenomena include patterns of colonisation, herding, and flocking. (The “Boids” model (62) is a classic example from artificial life in which realistic swarming behaviour is reproduced with a very simple, individual-based approach.) In addition to representing higher resolution dynamics than what can be included in a population-based model, individual-based models also facilitate the inclusion of a spatial component. Whereas population-based models are rarely spatially explicit, the organisms in individual-based models are invariably distributed about a heterogeneous landscape.

A large number of individual-based models have been written that differ widely with regards to the kinds of attributes used to define an organism, and in the choice and implementation of rules. A few of these models have been selected as illustrative samples, and will be reviewed here. Each of these was developed to represent phenomena that could not have been easily modelled using any other approach.

Spatially explicit behaviour in bird populations, whether it be the selection of nesting sites or the establishment of territorial areas, is a common focus of individual-based modelling efforts. Wolff (76), for example, used a spatially-explicit, individual-based approach to model the breeding and nesting behaviours of wood storks in the Florida everglades region. Each bird was modelled as a distinct individual, and the environment was represented by a grid of cells in which the water level and abundance of fish was varied according to seasonal patterns. The feeding requirements and social behaviour of the storks were described with simple rules. The model provided a fairly realistic description of the movement of the storks about their environment. Simulations based on the model were used to illustrate the importance of different individual behaviour patterns on overall population survival in a highly fragmented landscape. Another example of a fairly detailed individual-based model of a bird species is that of Letcher et al. (38) which was written to examine the territorial behaviour of the red-cockaded woodpecker in a spatially explicit setting. This model differs from other ones in that the landscape was not partitioned into a grid of fixed-sized cells. Instead, through the application of several behavioural rules describing the life cycle and habits of individual woodpeckers, the modelled birds established circular territories, the radii of which were allowed to vary within a continuous range of values.

The HOOFS model of Beecham and Farnsworth (3) depicts animal foraging behaviour based on a weighted consideration of both social factors and food availability. HOOFS was written as a general foraging model, in which animals are represented as generic objects, and energy is measured in an arbitrarily defined “food unit currency”. It has been used in simulation to explore the feeding behaviour of different types of herbivores, with an emphasis on large grazers. Simulations using HOOFS are event-driven, and each animal object is a separate thread that is maintained under the control of a central scheduler. Although current simulations are purely individual-based, the authors suggest that a “genetic” component could easily be incorporated as part of the animal object’s definition, allowing for the evolution of animal foraging strategies. Should this modification be added, HOOFS would then fall into the category of agent-based models described below.

Plants, as well, have been modelled with individual-based approaches, usually with the intent of studying forest succession or the development of canopy structure. The JABOWA models are perhaps the best known (5; 28). In these, young individual trees surrounding the area dominated by one fully grown tree compete for light and space. Trees are defined with attributes for age and size-specific traits, as well as environmental tolerances. The abiotic environment is described with an elevation, soil depth, moisture holding capacity and rockiness, temperature and precipitation rate. JABOWA-like models have been used widely to study succession in very different types of forest, and the dynamics obtained with them show good agreement with physical data.

Although in all of the models described above, interesting aspects of population dynamics are reproduced, in none of them are multi-trophic level systems that include both plants and animals depicted. In contrast, a group at Yale University (4) has developed one of the few reported individual-based multi-trophic ecosystem models (“Gecko”). In Gecko, the growth, movement, reproduction and feeding behaviour of individual organisms is modelled in a spatially explicit setting. The model does not represent organisms or the environment to a great degree of detail, and can, therefore, be easily tuned to represent quite different physical systems. So far, it has been calibrated to model two- and three-species grassland food webs with the intent to study more complex multi-species ecosystems in the future. The EcoCyborg ecosystem model (56) is another individual-based multi-species model that has the potential to include a wide variety of species types, ranging from microorganisms to all types of plants and animals. The model has been tested with configurations based on one- and two-trophic level food webs consisting of up to one dozen species. In these simulations, the modelled systems exhibited complex dynamics at many scales, including autopoiesis, self-organisation, and emergent spatial patterns.

The paucity of large-scale, multi-species individual-based ecosystem models is partly due to a number of implementation concerns, which have been outlined by McCauley et al. (45): (1) the computational intensity limits the number of individuals that can be simulated in a reasonable amount of time, and insufficient numbers may distort the population dynamics due to a lack of representation of some traits; (2) the non-linearity of the interactions, and the large number of rules, can make it difficult to discern the mechanisms responsible for population dynamics; and, (3) the execution of rules can be sequential or concurrent, which will result in significantly different dynamics for a given model, as will the order of rule execution in a sequential case. In addition, for sequential models, the relative impact of execution order is highly dependent upon the size of the simulation time step; although there are a number of measures that can be taken to mitigate this effect (e.g., 55). Also, for many individual-based models, the selection of appropriate attribute values can be difficult and very time-consuming (22). The consequence is that most individual-based models are written to represent species for which a lot of field data has been collected, leading to the creation of models that are very empirical and species-specific, rather than being general in their nature and application.

3.2 Agent-based models

In agent-based models, the basic units are objects that have the ability to learn about their environment and modify their behaviour accordingly. Thus, agent-based ecosystem models are similar to individual-based models, with the exception that organisms are provided with mechanisms by which they can adapt/learn/evolve, usually based on some type of genetic algorithm (although evolvable rule sets and artificial neural networks have also been used (70; 41)). While individual-based models tend to be fairly specific, usually being tuned so as to represent the behaviours and characteristics of real biological organisms, most agent-based models are more general, often having been written to explore theoretical questions regarding

the nature of biological evolution, or the role played by adaptation in the development of ecological communities. In addition, since evolution is considered to be a key factor in the dynamics elicited with these models, the organisms, or *agents*, are usually represented quite simply so as to permit simulations that execute over millions of life-cycles within a reasonable amount of real time.

There are two general types of agent-based models. In the first, an overriding genetic algorithm is used to explicitly select for certain traits or behaviours. In the second, illustrated below with examples such as *Tierra* and *EVOLVE III*, adaptive agents are created which reproduce and compete for resources. In these, the fittest variations survive, as determined by the modelled environment.

The modern version of the genetic algorithm is attributed to Holland (23; 24) and, in its simplest form, is as follows (50):

1. Start with a randomly generated initial population of M agents.
2. Allow the agents to live and reproduce for N generations and then calculate the fitness of each agent in the population (where fitness is assigned a numeric value calculated according to a specific function intended to select for a particular trait).
3. Repeat until the new population has M individuals: Select the fittest agents from the old population and allow them to reproduce to produce an offspring. Randomly mutate each locus in the offspring's genome. Place the offspring in the new population.
4. Go to step 2 with the new population.

Genetic algorithms are commonly used in the field of “evolutionary computation” to find solutions to computational problems (49). Since, for even a simple system, the number of possible evolved genomes is large, a good genetic algorithm must be efficient in evaluating the fitness of a given agent and then selecting variations, without compromising the current system state.

Holland has written a theoretical modelling framework called *ECHO* with which to investigate the selection process of genetic algorithms in systems based on ecological rules. In *ECHO*, agents are represented by digital “genomes” that determine their attributes and the types of interactions they can have with other agents. Each agent gathers resources from its environment, until it has enough resources to “reproduce”. Thus, the most successful agents will be those that are best at gathering resources in order to reproduce most frequently. During reproduction, the agent's genome is copied, and random mutation may occur. *ECHO* has been employed in a wide number of projects to explore the evolutionary process. The current release is written in C and is available on the Internet for public download.

Hraber and Milne (27) used a simplified version of *ECHO* (most stochastic mechanisms of evolution were disabled, and the number of allowable genotypes was constrained to 128) to explore the nature of community assembly in natural ecosystems. Results from experiments with a first model, in which agent interactions were based on genotypic specifications, and a second model, in which agent interactions were neutral and chosen randomly, showed that communities built upon the first model had stabilized at higher “productivity” levels than those based on the neutral model (where the number of different genotypes in a community is used as a measure of “productivity”). Hraber and Milne emphasised, however, that the applicability of their results to physical reality was not obvious, since within the *ECHO* framework there is no clear relationship between agents/resources and biological organisms in physical environments.

In the field of artificial life, one of the best known models of evolution is Ray's “*Tierra*” (58; 59;60; 61). Although it was not written to explicitly represent a natural ecosystem, a large

number of parallels can be drawn between the population dynamics that occur within “Tierra” and those of “real world” systems. During a simulation of “Tierra”, a single ancestor “organism” is created in the form of a program. The program’s instruction code (equivalent to the organism’s genome) provides it with the capacity to find free memory space in the RAM “soup” of its world and to produce a copy of itself. During the reproduction stage, there is the possibility for evolution through random mutations. The “soup” is quickly filled with the ancestor’s offspring, all of which must compete for memory (space) and CPU time (energy resources) in order to execute and reproduce. Thus, a highly competitive, Darwinian-style battle emerges in which only the fittest programs survive. In addition to providing interesting theoretical discussions about the nature of life, “Tierra” has been used extensively to study the mechanisms of natural evolutionary processes and the community level dynamics that emerge as a result.

EVOLVE III (54; 63) is one of the only agent-based models that is both genetically explicit and directly founded upon natural biology. In EVOLVE III, each individual organism is modelled with 15 phenotypic traits (e.g., temperature optimum, rate of energy intake, age) that are coded by a collection of up to 40 genes which are each represented as sequences of 200 nucleic acid bases. The model is event based; during the course of an organism’s life there are six types of events that may occur to initiate activities such as resource collection, reproduction, migration and death. The organisms interact on a two-dimensional spatial grid in which each cell has a unique set of environmental conditions such as light intensity and temperature. Many variants of the EVOLVE III model (containing up to 1000 organisms) have been implemented in simulation. In these, a number of evolved behaviours have been revealed, including symbiotic feeding, arms race development and adaptability to changing environmental conditions.

A widespread conclusion derived from simulations with agent-based models has been that an organism’s environment has a substantial influence on its behaviour and, subsequently, on the overall dynamics of the population of which that organism is a part. Many authors assert that the complexity of an evolving system, natural or artificial, is a reflection of the environment in which it resides (6). Thus, to evolve a highly complex, or sophisticated system, an environment should be provided that is variable, yet not so variable so as to kill the system. In agent-based models, for example, the contrived environment in which objects evolve must be variable, yet sufficiently regular for learning and adaptation to occur. Langton (36, p. 38) summarized this speculation in the statement: “Rigid, pre-specified, ‘unnatural’ environments foster rigid, predictable, ‘unlifelike’ evolutionary progression”. For this reason, in many object-based modelling efforts, detailed, spatially explicit environments are included. Often these are based on a cellular automata.

3.3 Cellular automata

In a cellular automata, the basic units for modelling (objects) are cells on a grid. A cellular automaton is a large tessellation of finite-state cells whose states are updated in discrete time steps according to deterministic rules, the operation of which depends on the states of neighbouring cells. Generally, the number of states that a cell can have is small (2-4) and the rules for determining them are quite straightforward. Nonetheless, as described by Wolfram (77), interesting results may emerge from seemingly simple configurations. Wolfram qualitatively defined four classes of characteristic limiting forms that a cellular automaton may attain and Langton (35) later embellished on these and provided a quantitative method by which to distinguish them. The four classes are as follows: (1) spatially homogeneous state [point attractor]; (2) sequence of simple stable or periodic structures [periodic attractor]; (3) chaotic aperiodic dynamics [strange attractor]; (4) complicated localized structures, some

propagating. The fourth class represents the state which Langton has coined “the edge of chaos”, and is the realm in which the dynamics of living systems is believed to fall.

Cellular automata have been used to model a number of spatial phenomena in ecology, particularly vegetation succession. For example, Hogeweg (21) randomly assigned hypothetical plant species (from 40 possibilities) to cells on a grid and used a probabilistic function to calculate each cell’s next state based on the frequency of species in the neighbouring cells. A small probability was reserved for the influx of a species not located nearby. Successive iterations resulted in the formation of different patterns in the vegetation map. Green (16; also described in: 21) used cellular automata to study the effects of “space-filling” processes (such as seed dispersal or animal migration) in contrast to “space-clearing” processes (fire, storms and other large disturbances). Through the use of a cellular automata model of an Australian forest, Green concluded that in the absence of space-clearing effects, vegetative clumping patterns promoted the continuance of species that would otherwise be eliminated by superior competitors. Consequences of this include the formation of ecological zones that help to maintain diversity in a forest and which are resistant to change. Green also showed that the introduction of clearing or fire to such a forest community caused sudden, catastrophic changes to its structure. In this work, the use of a cellular automaton model facilitated the representation of forests at a relatively high resolution, and enabled the effective illustration of how large scale features in a forest landscape can arise as a result of numerous fairly small scale disturbances.

Cellular automata have been used in many different landscape ecology applications. If a large area is being modelled at a high resolution, this approach can rapidly become computationally intensive. Gimblett et al. (14) describe a framework for a large watershed modelling project in which a GIS is used to store information about the state of cells on a landscape, and a CM-5 parallel computer is used to execute the model. They argue that this framework, in which the ability to store a large number of state variables for each cell is coupled with high performance computing, will enable the development of detailed landscape models that are based on historical satellite data.

Cellular automata are limited by the fact that they inherently represent spatial objects (although some research has been done to develop “mobile” cellular automata, in which a cell’s neighbours are not fixed (46)). Hogeweg (21) proposed the use of a non-synchronous, “poly-automata” model, the structure of which would allow for the inclusion of both individual organism and spatial objects. The more common solution, however, is to represent an ecosystem with a cellular automata-type model of an environment or terrain in combination with an individual or agent-based model to depict biological components.

4. Special programming methods and tools applicable to object-based ecosystem modelling

The models described above are all examples of representational modelling, in which an ecosystem is represented as a collection of interacting objects. Each representational model, whether it be individual-based, agent-based or a cellular automata, must be encoded in computational form in order to be implemented in simulation. The manner in which this is done will often have a large effect upon the model’s performance in simulation, and the selection of an appropriate language and simulation platform is always a key consideration in the development of any computer-based model. Naturally, many diverse languages and platforms were used to encode the different models discussed above. There is, however, one method that is used more often than others, and that is object-oriented programming.

While object-based models may be encoded in standard procedural programming languages, they are naturally suited to the object-oriented method. For this reason, there is a number of object-oriented computer languages, development environments and simulation platforms that have been created specifically to support the creation of object-based models. Object-oriented programming will be briefly described below, followed by descriptions of several simulation platforms that have been developed to facilitate the creation and testing of object-based ecosystem models. It should be noted here that “object-oriented” refers to a programming method, whereas “object-based” refers to a representational modelling approach.

4.1 Object-oriented programming

Although any model can be encoded in an object-oriented programming (OOP) language, object-based models are ideally suited to this method. The building blocks of an object-oriented program are taxonomic groups of related entities which are referred to as classes. Individual occurrences of classes are called instances. A key concept of OOP is encapsulation: both the data and the code associated with an instance are incorporated in its description. An instance’s data structure is defined with a collection of variables, or attributes. Instances of each class have sets of specific behaviours, or self-contained subroutines (code), which are called methods. Instances communicate with one another by sending messages which are interpreted by their methods. Since methods are encapsulated in instances, different types of instances can respond differently to the same message. This is known as polymorphism. All entities are organized into a hierarchy of classes and subclasses which inherit the attributes and methods of their superclasses. In the description of a subclass, specialised methods and attributes may be added, and the values that an inherited attribute can assume may be constrained. For more complete descriptions of object-oriented programming, see Rumbaugh et al. (64) and NeXT Software (53).

When encoding an ecosystem model in an OOP language, the most common method is to represent objects (individuals, agents, or cells) as instances of classes. Each object then has a set of unique attribute values, as well as a collection of behaviours that it shares with other objects of its type. Examples of ecosystem models based on the OOP formalism include those described by Maley and Caswell (42), the predator-prey model of Plant and Stone (57), and the animal foraging model of Beecham and Farnsworth (3). Of course, OOP is not limited to this one type of model: since OOP languages have a built-in object-based structure, it is relatively easy to encode any type of object-based representational model with them.

Despite the conceptual strengths of the object-oriented programming method, until recently, few ecosystem models had been written in OOP languages. Silvert (66) has suggested a number of reasons why this technology was not readily adopted by ecologists. Among the disadvantages he listed were the inefficiency of the original compilers of object-oriented languages, which made the development of large models infeasible due to the slow simulation execution time, and a lack of support for the use of floating-point numbers in early language versions. In addition, due to the nature of OOP languages, models encoded with this method are usually implemented as discrete event simulations (in which events to take place are queued in sequence, and time is updated accordingly), whereas most ecosystem models are implemented in time-driven simulations. Over the past few years, however, these drawbacks of object-oriented programming have been slowly overcome: object-oriented language compilers have become more efficient, computer hardware performance has improved, and a number of object-oriented simulation platforms have been developed specifically for the creation of object-based ecosystem models.

4.2 Object-based simulation platforms

One of the major concerns with regards to all types of object-based models is that the results of simulations based on them are virtually impossible to verify (i.e., it is difficult to determine whether or not the performance of the computational model is as intended, or if it is due to programming errors or other encoding mistakes). Murdoch et al. (51) discussed the issue of verification of simulations when using individual-based models, and gave an example of how a fairly detailed model could be tested through the selective isolation of different components. Although it is feasible, this process of error detection is, of course, time consuming and difficult. Thus, to bypass some of these difficulties, and to avoid the need for ecosystem modellers to keep “reinventing the wheel,” several groups have developed simulation platforms in which object-based computational models can be implemented. The two described in detail here, SWARM and ECOSIM, are both written in object-oriented programming languages.

The Swarm Simulation System was originally developed at the Santa Fe institute and is now maintained by the Swarm Development Group (69; 37; 48; 20). SWARM is a set of software tools written in Objective-C, an object-oriented extension to C. SWARM includes libraries of standard object design and creation routines, analysis tools, and a simulation kernel that supports hierarchical and parallel processing. It is specifically geared towards the simulation of agent-based models composed of large numbers of objects. Since all of the software is distributed free of charge, it can be used by anyone possessing the appropriate hardware to compile and run SWARM projects. Thus, its developers hope that the platform may serve as a standard test-bench for modelling complex systems, with which researchers can conduct repeatable experiments under known conditions. SWARM was not, however, written specifically for ecosystem modelling, and lacks a number of features that ecological modellers require, including the provision to integrate real data from GIS and similar databases. In addition, SWARM simulations are event-based, whereas most ecosystem models are more readily implemented as time-based simulations. For these reasons, ECOSIM (described below), as well as a number of other related systems (e.g., the Integrating Modeling Environment, 72), may be more appropriate modelling platforms for use by ecologists.

ECOSIM (39; 68) is another platform that has been developed to address some of the problems of code generation and verification encountered in object-based modelling. Like SWARM, the ECOSIM source code and documentation are distributed via the Internet free of charge. ECOSIM, however, has been written solely for the purpose of ecosystem modelling. It consists of a large number of C++ classes that facilitate the development and encoding of all three types of object-based models commonly used in ecology (i.e., individual-based models, cellular automata and genetic algorithms). Models that combine aspects of each type can also be created. The landscape, for example, can be represented with a cellular automata, and adaptable individuals can then exist within cells on the landscape. ECOSIM includes animation and visualization routines, as well as an event-based simulation kernel that is used to manage the activities of objects during a simulation.

Although SWARM and ECOSIM are probably the most extensive object-based simulation platforms that have been developed for public use, there are several other smaller platforms that can be useful for simple modelling exercises. Some of these are described in the review by Lorek and Sonnenschein (40). None, however, are sufficiently sophisticated or optimized for simulations with large numbers of objects. For example, INTER-SITE consists of a set of routines written in Python (an interpreted language) that have been developed specifically to simulate spatially realistic meta-population models (11). Although many such platforms exist, most are appropriate only for simple models that include small (<500) numbers of objects. Also, because these are written as general ecosystem simulation platforms, there is often a

considerable amount of revision required in order to tailor the code to meet the requirements of a particular modelling project. Thus, despite the existence of these various platforms, most object-based ecosystem models continue to be written “from the bottom-up”.

5. Conclusion

Through the representation of systems at a high resolution component level, simulations based on object-based computational models can, to some extent, emulate the massively parallel and highly distributed character of the natural world, and are well suited as research tools that can be used to provide insight into the origins of certain system dynamics. For this reason, there has been a general trend in ecosystem modelling away from conventional models (i.e., models in which population dynamics are described with sets of continuous differential equations) in favour of the three types of object-based models described above. There still remain, however, a number of excellent reasons to use a conventional modelling approach rather than an object-based one. For example, although mathematical formulations of population dynamics are not based on representations of lower level components whose activities cause higher level change, simulations with these may still provide a good portrayal of a population’s variation and, under some conditions (e.g., spatial homogeneity), their predictions may very well match those of an equivalent object-based approach (45; 75). Also, simulations based on conventional models are usually far less computationally intensive than their high-resolution object-based counterparts, and their results are much easier to verify. In general, the selection of one modelling approach over another must be made in the context of an overall research objective. If the purpose is to explore the complex characteristics of an ecosystem’s structure and dynamics, then an object-based approach is an obvious choice.

To date, few large-scale, object-based ecosystem models with many components have been created, partly due to the concerns discussed above with regards to the specification of appropriate attribute values to describe the states of all the objects, and also due to the computational effort required to implement an object-based model in simulation. Instead, most of the object-based models that have been developed are either: a) single-species models that are strongly empirically based, or, b) abstract, general models that illustrate theoretical principles but lack real-world descriptive capacity. Although the specification of attribute values will always be a concern that will make it difficult to apply the object-based approach in the development of large-scale ecosystem models, increased computing capacity is rapidly alleviating some of the other drawbacks. The result is that object-based modelling is becoming more feasible, and may be used more commonly for large ecosystem modelling projects in the future.

The adoption of new modelling techniques as a means of incorporating complex system studies into ecology is coincident with a growing belief that ecosystem management practices must become more integrative and flexible (26). It is for this reason that object-based modelling approaches are being considered for large-scale environmental assessment projects. The compartment of an ecosystem is fundamentally complex, and cannot be accurately predicted by models that are constrained to a narrow range of variables (25; 29). Appropriate management policies cannot be based on solutions derived from knowledge of only small parts of a larger, integrated system, nor can they be based on assumptions of constancy and stability in dynamic relationships. Increasingly, the compartment of an ecosystem is shown to arise as the result of a multiplicity of non-linear, inter-related causes that span multiple levels of organisation. Although global features such as species persistence or patterns of vegetation distribution on a landscape are seen to emerge as the result of self-organisation, they are never

repeated exactly, and they may dissolve at any time, leaving a gap to be filled by a new, emergent feature.

A common criticism of object-based models is that they are difficult to validate (i.e., to determine to what extent each rule in the model contributes to the dynamics that arise in different simulation scenarios, and to assess whether these rules are based on valid assumptions). This is of particular concern since simulations based on object-based models often produce complex results! A second criticism is that they are rarely calibrated, or integrated, with data from real systems, which is for the most part untrue: The majority of the individual-based and cellular automata models discussed here have in fact been calibrated against real data sets. Of course, validation and calibration are generally performed synchronously, and those models which have been calibrated with real data have usually undergone extensive validation in the process. There is a large number of models, particularly individual-based plant models, that have been developed and tuned over many years to represent a wide variety of data sets (e.g., 10; 7) from different forests growing under varied environmental conditions.

There is an increasing number of object-based models that have been profitably applied to ecosystem management scenarios. These are often hybrid systems, in which data from a GIS or other database is linked to different simulation modules. In such cases, the principal components represented by objects in the model are landscape units. Examples include Maxwell and Constanza's "open geographic modeling environment" (44) and the visually oriented forest land use program developed by Fordham et al. (9). Not all management tools are landscape based, however. Gimblett et al. (13) addressed the problem of forest recreation management with a model in which individual hikers, cyclists, and jeep drivers were the principal components. This model was designed to help park managers develop trail systems that most effectively eliminated conflicts between these three main types of recreationists in a national forest area. In all of these models, the conclusions and decisions drawn from the simulations are those that could not be reached without the use of an object-based model, in which spatial components and detailed interactions at the object-level determine the outcome of a simulation.

Due to the fundamental unpredictability of an ecosystem's compartment (12), simulations using object-based computational models cannot be used to precisely forecast events over the long-term. They may, however, contribute to the fields of natural resource management and ecology by providing useful illustrations of the nature of ecosystem dynamics and of the mechanisms that give rise to unexpected events. Thus, by facilitating the incorporation of interactions between ecological, economic and social components, such models may provide resource managers, biosystems engineers, and others with a clearer understanding of which events are indicative of impending catastrophe, and which are simply a part of the normal envelope of dynamics of the system under study.

References

- 1 Ashby, W.R. (1956) *An Introduction to Cybernetics*, Chapman & Hall, London.
- 2 Assad, A. M. and Packard, N.H. (1992) Emergent colonization in an artificial ecology, in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, (ed.) F.J. Varela and P. Bourguine, 143-152, MIT Press, Boston.
- 3 Beecham, J.A. and Farnsworth, K.D. (1998) Animal foraging from an individual perspective: an object orientated model, *Ecological Modelling*, 113: 141-156.
- 4 Booth, G. (1997) Gecko: A continuous 2-D world for ecological modelling, *Artificial Life*, 3(3): 147-163.
- 5 Botkin, D. B., Janak, J. F. and Wallis, J. R. (1972) Some ecological consequences of a computer model of forest growth, *Journal of Ecology*, 60: 849-872.
- 6 Bonabeau, E. W. and Theraulaz, G. (1994) Why do we need artificial life? *Artificial Life*, 1(3): 303-325.
- 7 Chertov, O., Komarov, A. and Tsiplianovsky, A. (1999) A combined simulation model of Scots pine, Norway spruce and Silver birch ecosystems in the European boreal zone. *Forest Ecology and Management*, 116: 189-206.
- 8 Fahse, L., Wissel, C. and Grimm, V. (1998) Reconciling classical and individual-based approaches in theoretical population ecology: A protocol for extracting population parameters from individual-based models. *The American Naturalist*, 152(6): 838-852.
- 9 Fordham, D.P., Hood, L.M. and Malafant, K.W. (1997) Complexity in landscapes and resource planning: packaging science for decision makers. 1997 Proceedings of the Ecological Society of Australia, Chapter 22.
- 10 Friend, A.D, Stevens, A.K., Knox, R.G. and Cannell, M.G. (1997). A process-based, terrestrial biosphere model of ecosystem dynamics (Hybrid v3.0). *Ecological Modelling*, 95: 249-287.
- 11 Gathmann, F.O. and Williams, D.D. (1998) Inter-site: a new tool for the simulation of spatially realistic population dynamics, *Ecological Modelling*, 113: 125-139.
- 12 Gell-Man, M. (1994) *The Quark and the Jaguar*, W.H. Freeman & Co., New York.
- 13 Gimblett, R., Richards, M., Itami, R. (1998) A complex systems approach to simulating human behaviour using synthetic landscapes, *Complexity International*, 6.
- 14 Gimblett, R., Ball, G., Lopes, V. et al. (1995) Massively parallel simulations of complex, large scale, high resolution ecosystem models, *Complexity International*, 2.
- 15 Green, D. G. (1994) Connectivity and complexity in landscapes and ecosystems. *Pacific Conservation Biology*, 1(3): 194-200.
- 16 Green, D. G. (1993) Emergent behaviour in biological systems, in *Complex Systems - From Biology to Computation*, (eds.) D.G. Green and T.J. Bossomaier, 24-35. IOS Press, Amsterdam. (Also published in Vol. 1 of *Complexity International*.)

- 17 Grimm, V. (1999) Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling*, 115: 129-148.
- 18 Haken, H. (1977) *Synergetics: An Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology*. Springer-Verlag, Berlin.
- 19 Heylighen, F. (1997) Publications on complex, evolving systems: A citation-based survey, *Complexity*, 2(5): 31-36.
- 20 Hiebeler, D. (1994) The Swarm simulation system and individual-based modeling, in *Proc. of Decision Support 2001: Advanced Technology for Natural Resource Management*, Toronto, ON, Sept. 1994.
- 21 Hogeweg, P. (1988) Cellular automata as a paradigm for ecological modeling, *Applied Mathematics and Computing*, 27: 81-100.
- 22 Hogeweg, P. and Hesper, B. (1990) Individual-oriented modelling in ecology, *Mathematical and Computational Modelling*, 13(6): 83-90.
- 23 Holland, John H. (1995) *Hidden Order: How adaptation breeds complexity*, Addison-Wesley Publishing Co., Reading, MA.
- 24 Holland, John H. (1975) *Adaptation in Natural and Artificial Systems. An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Ann Arbor, MI
- 25 Holling, C.S. (1996) Engineering resilience versus ecological resilience, in *Engineering within Ecological Constraints*, (ed.) P. Schulze, 31-43, National Academy Press, Washington, DC.
- 26 Holling, C.S. (1993) Investing in research for sustainability, *Ecological Applications*, 3(4): 552-555.
- 27 Hraber, P.T. and Milne, B.T. (1997) Community assembly in a model ecosystem, *Ecological Modelling*, 103: 267-285.
- 28 Huston, M. (1992) Individual-based forest succession models and the theory of plant competition, in *Individual-Based Models and Approaches in Ecology*, (eds.) D. DeAngelis and L. Gross, 408-420, Routledge, Chapman and Hall, Inc., New York.
- 29 Jørgensen, S, Mejer, H. and Nielsen, S. (1998) Ecosystem as self-organizing critical systems, *Ecological Modelling*, 111: 261-268.
- 30 Judson, O. (1994) The rise of the individual-based model in ecology, *Trends in Ecology and Evolution*, 9(1): 9-14.
- 31 Johnson, C.R. (1997) Self-organising in spatial competition systems. 1997 Proceedings of the Ecological Society of Australia, Chapter 20.
- 32 Kauffman, S. 1993. *The Origins of Order: Self-organisation and Selection in Evolution*. Oxford University Press, New York.
- 33 Kawata, M. and Toquenaga, Y. (1994) From artificial individuals to global patterns, *Trends in Ecology and Evolution*, 9(11): 417-421.
- 34 Klomp, N. and Green, D. (1996) Complexity and connectivity in ecosystems, *Complexity International*, 3.

- 35 Langton, C. (1990) Computation at the edge of chaos: phase transitions and emergent computation, *Physica D*, 42: 12-37.
- 36 Langton, C. (1989) Artificial Life, in *Artificial Life*, (ed.) C. Langton, 1-47, Addison-Wesley, Redwood City, CA.
- 37 Langton, C., Minar, N., Burkhart, R., Askenazi, M. and Ropella, G. (1995-99) The Swarm Simulation System. Web-based documentation: URL: <http://www.swarm.org>, Santa Fe Institute, Santa Fe.
- 38 Letcher, B. H., Priddy, H. A., Walters, J. R., and Crowder, L. B. (1998) An individual-based, spatially-explicit simulation model of the population dynamics of the endangered red-cockaded woodpecker, *Picoides borealis*, *Biological Conservation*, 86: 1-14.
- 39 Lorek, H. and Sonnenschein, M. (1998) Object-oriented support for modelling and simulation of individual-oriented ecological models, *Ecological Modelling*, 108: 77-96.
- 40 Lorek, H. and Sonnenschein, M. (1999) Modelling and simulation software to support individual-based ecological modelling, *Ecological Modelling*, 115: 199-216.
- 41 Maes, P. (1994) Modeling adaptive autonomous agents, *Artificial Life*, 1(1/2): 135-162.
- 42 Maley, C. C. and Caswell, H. (1993) Implementing i-state configuration models for population dynamics: an object-oriented programming approach, *Ecological Modelling*, 68: 75-89.
- 43 Maturana, H.R. and Varela, F. (1980) *Autopoiesis: The organization of the living*, D. Reidel Publishing Co., Dordrecht, Holland.
- 44 Maxwell, T. and Constanza, R. (1997) An open geographic modeling environment. *Simulation Journal*, 68(3): 175-185.
- 45 McCauley, Edward, Wilson, William G., and de Roos, Andre M. (1993) Dynamics of age-structured and spatially structured predator-prey interactions: Individual-based models and population-level formulations, *The American Naturalist*, 142(3): 412-442.
- 46 Miramontes, O., Solé, R., and Goodwin, B. (1993) Collective behaviour of random-activated mobile cellular automata, *Physica D* 63: 145-160.
- 47 Millonas, M.M. (1994) Swarms, Phase Transitions, and Collective Intelligence, in *Artificial Life III*, (ed.) C. Langton, 417-445, Addison-Wesley, Reading, MA.
- 48 Minar, N., Burkhart, R., Langton, C. and Askenazi, M. (1996) The Swarm Simulation System: A toolkit for building multi-agent simulations. <http://www.santafe.edu/projects/swarm/> June 21, 1996. Santa Fe Institute, Santa Fe.
- 49 Mitchell, M. (2000) *Life and evolution in computers*, Working paper, Biophysics Group, Los Alamos National Laboratory.
- 50 Mitchell, M. and Forrest, S. (1994) Genetic Algorithms and Artificial Life, *Artificial Life*, 1(3): 267-289.
- 51 Murdoch, W. W., McCauley, E., Nisbet, R. M., Gurney, W. S. C. and de Roos, A. M. (1992) Individual-based models: Combining testability and generality, in *Individual-Based Models and Approaches in Ecology*, (eds.) D. DeAngelis and L. Gross, 18-35, Routledge, Chapman and Hall, Inc., New York.

- 52 NECSI. (1999) New England Complex Systems Institute Web Site. URL: <http://www.necsi.org>, NECSI, New Hampshire.
- 53 NeXT Software Inc. (1996) Object Oriented Programming and the Objective-C Language, Next Software Inc.
- 54 O'Callaghan, J. and Conrad, M. (1992) Symbiotic interactions in the EVOLVE III ecosystem model, *Biosystems*, 26: 199-209.
- 55 Palmer, J. B. (1992) Hierarchical and concurrent individual based modeling, in *Individual-Based Models and Approaches in Ecology*, (eds.) D. DeAngelis and L. Gross, 189-207, Routledge, Chapman and Hall, Inc., New York.
- 56 Parrott, L. and Kok, R. (2000) Use of an object-based model to represent complex features of ecosystems. Presented at the Third Intl. Conference on Complex Systems, May 21-26, 2000, Nashua, NH, USA. InterJournal Manuscript #371.
- 57 Plant, R. and Stone, N. (1991) Object-oriented models, in *Knowledge-based Systems in Agriculture*, (ed.) R. Plant, 259-285, McGraw Hill, New York.
- 58 Ray, T. S. (1994a) TIERRA.DOC: Documentation for the Tierra Simulator V4.1.
- 59 Ray, T. S. (1994b) An evolutionary approach to synthetic biology: Zen and the art of creating life, *Artificial Life*, 1(1/2): 179-209.
- 60 Ray, T. S. (1994c) Evolution, complexity, entropy and artificial reality, *Physica D*, 75: 239-263.
- 61 Ray, T. S. (1992) Evolution, ecology and optimization of digital organisms, Santa Fe Institute working paper 92-08-042.
- 62 Reynolds, C. W. (1987) Flocks, herds, and schools: A distributed behavioral model (Proceedings of SIGGRAPH '87), *Computer Graphics*, 21(4): 25-34.
- 63 Rizki, M. M. and Conrad, M. (1985) EVOLVE III: a discrete events model of an evolutionary ecosystem, *Biosystems*, 18: 121-133.
- 64 Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorenzen, W. (1991) *Object-Oriented Modeling and Design*, Prentice Hall, New Jersey.
- 65 Santa Fe Institute. (1999) SFI Bulletin, 14(2), Santa Fe Institute, New Mexico.
- 66 Silvert, W. (1993) Object-oriented ecosystem modelling, *Ecological Modelling*, 68: 91-118.
- 67 Sole, R.V. and Bascompte, J. (1998) Emergent phenomena in spatially extended model ecosystems. In *Modeling Spatiotemporal Dynamics in Ecology*, (eds.) R.V. Sole and J. Bascompte, 1-25, Springer-Verlag, Berlin.
- 68 Sonnenschein, M., Köster, F., Lorek, H. and Vogel, U. (1994-1999) ECOSIM Web-based documentation, URL: http://www.offis.uni-oldenburg.de/projekte/ecotools/project_ecotools3.htm
- 69 Swarm Development Group. (1999) The Swarm Simulation System, Web-based documentation, URL: <http://www.swarm.org>

- 70 Taylor, C., Jefferson, D., Turner, S. and Goldman, S. (1988) RAM: Artificial life for the exploration of complex biological systems, in *Artificial Life*, (ed.) C. Langton, 276-295, Addison-Wesley, Reading, MA.
- 71 Taylor, R. A. J. (1992) Simulating populations obeying Taylor's Power Law, in *Individual-Based Models and Approaches in Ecology*, (eds.) D. DeAngelis and L. Gross, 295-311, Routledge, Chapman and Hall, Inc., New York.
- 72 Villa, F. (2000) Integrating Modelling Toolkit. Institute for Ecological Economics, University of Maryland, USA.
- 73 Waldrop, M. (1992) *Complexity*, Touchstone Books, New York.
- 74 Watts, D. and Strogatz, S. (1998) Collective dynamics of 'small-world' networks, *Nature*, 393: 440-442.
- 75 Wilson, W. G. (1998) Resolving discrepancies between deterministic population models and individual-based simulations, *The American Naturalist*, 151(2): 116-134.
- 76 Wolff, W. F. (1994) An individual-oriented model of a wading bird nesting colony, *Ecological Modelling*, 72: 75-114.
- 77 Wolfram, S. (1984) Cellular automata as models of complexity, *Nature*, 311: 419-426.

Endnotes

1. A complex system is very often a "small-world" network: a distributed system in which most components are connected locally in clusters, but in which there are a few 'random shortcuts' linking distant vertices, so that every component is connected to every other via a relatively short path (74).