# Learning to Engineer Life:
## Development of a generally configurable model for the simulation of artificial ecosystems.

by
Lael Parrott

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements of the degree of
Doctor of Philosophy.

February 2000.

Department of Agricultural & Biosystems Engineering
Macdonald Campus, McGill University
Montreal, Canada

# Abstract

The objective of this project has been to create a generally configurable computational model that can be used to simulate the behaviour of ecosystems. It was undertaken within the "EcoCyborg Project", a research program for which the long term goal is to learn how to engineer living systems, specifically *ecocyborgs*: ecosystems combined with technological control components. The short term goal of this research program is to improve the practice of ecosystem engineering by creating modelling and simulation tools with which specific design and management questions can be explored.

The principal intent in creating the model was to reproduce, in the virtual realm, some of the complex features and dynamics common to all large ecological assemblages. The model is constructed of simple, object-based representations of most of the major biotic and abiotic components that are found in any ecosystem, including a spatially explicit terrain, an atmosphere, and various plants and animals of different species. The behaviour of many of these components is driven by climate-related external forcing functions. Each component is represented by an object, the state of which is described by properties, and the functionality of which is described with rule-based expressions. During simulation, the state of each object is computed at regular time increments, and the system's global-level comportment arises as the aggregation of object-level events. As part of the thesis work, a complete set of configuration programs was written, with which the constitution, initial state, and climate of an ecosystem can be specified for simulation. The model is a novel contribution, due to both the degree of configurability provided by these programs, and the level of detail and breadth of scope with which an ecosystem can be represented.

The model was configured to represent a hypothetical, materially-closed ecosystem contained within a space station, and was then tested in simulation.

This modelled ecosystem exhibited complex behaviour similar to that of natural ecosystems, such as spatial and temporal self-organisation. The results obtained demonstrate the flexibility and potential utility of the model for use in both ecosystem engineering applications and the theoretical study of the underlying relationships between the constitution and comportment of ecosystems.

# Résumé

Ce projet avait pour objectif la création d'un modèle informatique configurable qui peut être utile à la simulation du comportement d' écosystèmes. Cette entreprise est partie intégrale du "EcoCyborg Project", un programme de recherche dans lequel on étudie l'ingénierie des systèmes vivants, plus spécifiquement d'*écocyborgs*: écosystèmes combinés à des mécanismes de contrôle technologiques. L'objectif à court terme de ce programme de recherche est l'amélioration des pratiques d'ingénierie d' écosystèmes par le développement d'outils de modélisation et de simulation grâce auxquels il sera possible d'explorer divers designs et méthodes de gestion spécifiques.

Le développement du modèle avait pour objectif principal la reproduction, dans le domaine virtuel, de certains des attributs et comportements complexes communs à tous les assemblages écologiques d'importance. Le modèle est construit à l'aide d'objets configurables simples, représentant la plupart des composants vivants et non-vivants retrouvés dans tout écosystème, ainsi qu'une description spatiale explicite du terrain, une atmosphère, des plantes et des animaux d'espèces diverses. Le comportement de la plupart de ces composants est gouverné par des fonctions contraignantes externes de conditions climatiques. Chaque composant est représenté par un objet, dont l'état est décrit par des propriétés, et dont la fonctionnalité est décrite par des expressions à base de règles. Durant la simulation, l'état de chaque objet est évalué à des intervalles de temps réguliers et le comportement global du système émerge de l'agrégation du comportement et de l'interaction de ces objets. Une collection complète de programmes de configuration a aussi été développée, à l'aide desquels la constitution, l'état initial, et le climat propre à un écosystème peuvent être spécifiés pour fin de simulation. Le modèle présenté est une contribution authentique, autant au niveau du degré de configurabilité possible, qu'au niveau des détails et de l'étendue avec lesquels on peut représenter un écosystème.

Le modèle à été configuré pour représenter un écosystème hypothétique matériellement fermé, contenu dans une station spatiale et ensuite testé en simulation. Les résultats de la modélisation de cet écosystème démontrent des comportements complexes similaires à ceux des écosystèmes naturels, tel l'auto-organisation spatiale et temporelle. Ces résultats démontrent aussi la flexibilité et l'applicabilité du modèle pour l'ingénierie d'écosystèmes et l'étude théorique des relations fondamentales entre la constitution et le comportement des écosystèmes.

## Acknowledgements

I gratefully acknowledge my thesis supervisor, Dr. R. Kok, for all of his support and advice over the years. He has spent countless hours of his time reading, writing and discussing ideas with all of us in the EcoCyborg Project. I'd like to personally thank him for all of the contributions he has made towards this work.

A number of other people also assisted in the completion of this project. I owe a warm and special thanks to Dr. J.A. Landry for allowing his computers to be taken over by the rabbits and voles, for assisting in the creation of the CD-ROM appendices, and for translating the thesis abstract. I'd also like to thank all of the other EcoCyborg Project members: Grant, Jennifer, Robert and Steve, for their friendly advice and input over the years. Special thanks to Jennifer for reading through the final draft of the thesis.

I'd like to acknowledge NSERC and McGill University for providing the funding required to complete this project, and the Department of Agricultural and Biosystems Engineering for all of its support.

Finally, I'd like to thank all of my friends and family members, especially Mom, Dad, Tatiana, Dave and Linda, for their long-distance patience and support over the years.

# Guidelines for a Manuscript Based Thesis

This thesis has been prepared in accordance with the February, 1999 revision of the Guidelines for Thesis Preparation (Faculty of Graduate Studies and Research, McGill University). It is stated therein that:

*"As an alternative to the traditional thesis format, the dissertation can consist of a collection of papers that have a cohesive, unitary character making them a report of a single program of research. [...] The thesis must be more than a collection of manuscripts. All components must be integrated into a cohesive unit with a logical progression from one chapter to the next. In order to ensure that the thesis has continuity, connecting texts that provide logical bridges between the different papers are mandatory.*

*In general, when coauthored papers are included in a thesis the candidate must have made a substantial contribution to all papers included in the thesis. In addition, the candidate is required to make an explicit statement in the thesis as to who contributed to such work and to what extent. This statement should appear in a single section entitled 'Contributions of Authors' as a preface to the thesis."*

## Contributions of Authors

In accordance with the McGill "Guidelines for a Manuscript Based Thesis", the contributions of the candidate and the co-authors towards the completion of this thesis work are set down here.

The conceptualization, development, and creation of the ecosystem model, as well as the completion of all of the experimental simulation work, was the sole responsibility of the candidate. The C language code for the ecosystem model was written entirely and solely by the candidate and can be found on the included CD-ROM (Appendix B). The model has been developed in conjunction with a number of Basic language "specification" and "definition" programs that are used to configure an ecosystem starting state, and to set simulation parameters. The main routines for these programs were also written by the candidate (see CD-ROM). The user interface subroutines for these programs were originally written by Dr. Robert Kok and were later modified by Dr. Grant Clark.

Chapters 2 through 6 of this thesis are coauthored by the candidate and the thesis supervisor, Dr. Robert Kok. These will serve as the working texts for manuscripts to be submitted for publication. The original documentation and the creation of associated diagrams for these manuscripts was the responsibility of the candidate. The co-author guided the candidate towards depth and clarity in the presentation of material, and thoroughly critiqued and reviewed the content of each manuscript. Appendix C was coauthored by the candidate and Drs. Grant Clark and Robert Kok, with equal contributions by all of the authors. All other chapters of the thesis were authored by the candidate.

# Table of Contents

## List of Figures

## List of Tables

# About the CD-ROM

Appendices B, C and D are located on the enclosed CD-ROM.

*To view the contents of the CD:*

If AutoRun is enabled for your CD-ROM drive, just insert the CD and the main page of the appendices will be automatically loaded by your default web browser.

If AutoRun is not enabled, insert the CD into your CD-ROM drive and then in Windows Explorer, double-click on the file /WWW/index.htm on the CD.

*Software requirements:*

This CD has been created for computers running Windows 95, 98, NT or 2000.

A frames-compatible web browser is required, and Javascript support is recommended. For your convenience, Internet Explorer v.5.0.1 for Windows is included on the CD-ROM in the directory: Internet Explorer 5.0.1 (32-bit for Win 95-98-NT).

Adobe Acrobat is required to view Appendix C. The Windows 98 version of this software is included on the CD-ROM in the directory: Acrobat. Other versions can be obtained from: http://www.adobe.com.

The material is best viewed at a resolution of 800x600 pixels.

## Chapter 1: Learning to Engineer Life: An Introduction

The research described herein is an exploration of the nature of living systems - their patterns of organisation, structure, and dynamic processes - with the aim of gleaning some ideas about the manner in which they might be engineered. It is being pursued as part of a research program called the EcoCyborg Project[1], the long-term objective of which is to learn how to engineer all types of biosystems, ranging from artificial minds to insect colonies, enclosed agro-ecosystems and extra-terrestrial life-support systems. It is, therefore, an exceptionally multidisciplinary research program, in which concepts from a variety of fields, including artificial life, artificial intelligence, ecology, engineering, and the space sciences, are integrated. Present work is focused on an investigation of complex biological networks such as ecosystems, that are guided by, or *cyborged* with, intelligent control systems. Such systems are called "ecocyborgs", in recognition of the merger of ecological and cybernetic influences in their design.

This research program was begun, in part, to address the circumstance that, although many types of ecological systems are currently being *constructed* by engineers, there is no general set of theoretical *design* principles according to which such a task may be approached. Most ecosystem engineering (i.e., design, construction, operation, repair, and maintenance), as it is currently performed, is based on experiential knowledge and trial-and-error experimentation. In the near future, however, human population pressures on Earth are likely to lead to an increasing need for ecosystem engineering, in order to manage and maintain the health of the ecosystems in which humans reside, as well as for the restoration and preservation of the remaining wilderness on this planet. Space-based travel and inhabitation projects will also require expertise in the engineering of ecosystems to support terrestrial life in places other than Earth. In order to accomplish all of this, we will need to have a better understanding of the relationships between ecosystem structure and composition, and the complex

---

[1] Further information regarding the rationale and objectives of the EcoCyborg Project is given in Appendix C, on the enclosed CD-ROM.

dynamics that emerge from these relationships. A goal of the EcoCyborg Project, therefore, is to develop tools with which we can begin to understand these relationships, thereby refining the practice of ecosystem engineering.

In consideration of the potential applications of ecosystem engineering to the space sciences, the case that is of particular interest is a materially-closed ecocyborg that is resident within a cylindrical space station. Although this system is purely hypothetical, it is based on plausible near-future technology, and serves as a convenient case study. In the present phase of the project, a computational model of the space-station ecocyborg is being created and implemented in simulation in order to explore some of the aspects that need to be considered when engineering such a system. This model is composed of several parts which are being developed independently by different members of the EcoCyborg Project research group. The parts include an ecosystem, Pavlovian and cognitive level control mechanisms, and a number of "forcing functions". The major contribution made to the EcoCyborg Project with this thesis work has been the development of a model with which the ecosystem and forcing function parts of the space-station ecocyborg can be represented and simulated.

The objective of this work was, therefore, to create a generally configurable computational model that could be used to simulate the behaviour of ecosystems. The model was not intended to specifically represent, describe, or predict the behaviour of any particular ecosystem but, rather, was developed with the intent of reproducing, in the virtual realm, some of the complex features common to all large ecological assemblages. The specific requirements with regards to the scope of the model and its implementation in simulation are presented in detail in Chapter 3, Section 3.3. In this chapter, some of the terminology and speculative research questions underlying the development of the ecosystem model are presented.

## 1.1. Complex system studies

This work was undertaken in the context of complex system studies, in which an ecosystem is generally recognized to be a "complex system", which is defined in Chapter 2 of this thesis as: "a small-world network of many components whose aggregate behaviour is both due to, and gives rise to, multiple-scale structural and dynamical patterns which are not inferrable from a system description that spans only a narrow window of resolution". A key aspect of this definition is the concept of scale, or the "lens" through which a system is observed, since, in a complex system, different components and structures are evident only at different spatial and temporal resolutions. Generally, these "observable regularities" cannot be easily described with a few parameters, or even with a single model. Thus, much of the research pertaining to complex systems has been focused on the elucidation of effective measures with which the *complexity* of a system can be characterized.

### *1.1.1. Complexity*

The complexity of a system is often defined simply as a measure of the amount of information required to describe it. Thus, Gell-Man (1994, p.34) has defined the crude complexity of a system to be the "length of the shortest message that will describe a system, at a given level of coarse graining, to someone at a distance, employing language, knowledge and understanding that both parties share (and know they share) beforehand". Similarly, in the field of computer science, the algorithmic complexity, or algorithmic information content, is the length of the shortest program that will print out a bit string that completely describes a system (Solomonoff, 1964). A more descriptive measure of complexity, however, is one that characterizes only the regularities in a system. Thus, Gell-Man has proposed the concept of effective complexity, to refer to the length of the schema that is used to describe just the regularities in a system. Inherent in this definition is the acknowledgement that all perceived regularities are defined relative to an observing system, which may itself be highly complex. The perceived complexity of a system, therefore, will vary depending upon the observer. Similarly, in the context of the EcoCyborg Project, Clark (1999) has defined the degree of complexity of a system as being a measurement of how difficult it is for an observer to describe the identifiable regularities in its overall constitution and comportment.

It may be that there is no single, effective measure with which the complexity of a system can be established. Acknowledging the difficulty of assigning a single value to complexity, Wolpert and Macready (1997) proposed that the complexity of a system can be characterised by a "self-dissimilarity" signature that is a vectorised measure of the differences in structure between scales (technically, the amount of information inferrable from the structure at one resolution that goes beyond what was inferrable from the structure at a finer resolution). They proposed a function with which the "distance between probability distributions" at different resolutions could be measured. Similarly, Bar-Yam (1999) has suggested that systems can be characterised by a "complexity profile" that is a cumulative sum of the number of different behaviours exhibited by a system at different scales.

### *1.1.2. Features of complex systems*

The systems that are generally acknowledged to be complex (e.g., minds, societies, and biological organisms) are made up of large collections of interacting components whose composite behaviour gives rise to patterns in higher-level structures and dynamics, without the presence of any central influence. When such behaviour is unexpected, that is, when it is not immediately foreseeable upon inspection of the specification of the system, it is termed *emergent*. More precisely, emergence may be defined as "a process in which a collection of interacting units acquires qualitatively new properties that cannot be reduced to a simple superposition of individual contributions" (Prusinkiewicz, 1994, p. 61). Emergence gives rise to some of the most interesting dynamics in complex systems, including the formation of waves and vortexes, the social ordering of bee colonies, and the development of consciousness in brains. Assad and Packard (1992, p. 145) have defined emergence on a relative scale from weakly emergent ("behavior is deducible in hindsight from the specification *after* observing the behavior") to strongly emergent ("behavior is deducible in theory, but its elucidation is prohibitively difficult") and maximally emergent ("behavior is impossible to deduce from the specification"). They hypothesized that strong or maximal emergence seems to be a typical characteristic of the dynamics of most living systems.

Emergence is the product of *self-organisation*: the generation of spatio-temporal order in the absence of any macroscopic description of that order[1]. Usually, self-organisation is achieved and maintained by a dynamic system so long as there is a continual flow of some resource (e.g., energy, matter, or information) across the system boundary. The result is a type of dynamics that falls somewhere between the extremely ordered (e.g., simple periodic cycling) and the totally disordered (e.g., uncorrelated random noise), and a type of structure that reflects an intermediate degree of connectivity between components.

Thus, a complex system is typically a large network of interrelated entities whose activities, in combination, cause sustained, substantially ordered structure and comportment at the system, or global, level. Unlike a fractal (Mandelbrot, 1982), which is self-similar across many scales, a complex system is dissimilar at different scales. From an information-theoretic perspective, it can be argued that a complex system exploits the capacity to process information by differentiating its processing ability across scales through the use of an organisational structure that is markedly dissimilar at different resolutions. In this regard, Wolpert and Macready (1997) have hypothesized that in nature there is a selective advantage to maximising the amount of information processed by a system in any given volume of space, since a system that processed information similarly on every scale would not be making efficient use of its available resources. Similarly, Holling (1996) has proposed that natural mechanisms of management and

---

[1] The term, self-organisation comes from the work of Haken (1977) in synergetics. Haken used the term to describe a phenomenon in laser physics in which a collection of electromagnetic waves of different wavelengths form, under certain conditions, a coherent wave of a certain wavelength.

self-regulation in ecosystems maintain these systems on the "edge of instability" in order to maximise information processing, and, therefore, evolutionary opportunity.

### *1.1.3. Ecosystems as complex systems*

The concepts of emergence and self-organisation are becoming increasingly prevalent in ecology, where they are used to explain a number of phenomena in ecosystems. In these systems, the seemingly random, or erratic, behaviour of individual organisms, in conjunction with random environmental influences, has been shown to lead to 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) which, in turn, affect the behaviour of the individuals in the populations. Commenting on this, Fahse et al. (1998, p.849) concluded that: "It may turn out in the future that the notion of ecological systems as being self-organised leads to a deeper understanding of ecological principles than the notion of self-regulation and negative feedback, which are cybernetic notions and as such omit any reference to the basic unit of ecological systems, the individuals." A related concept is that of *autopoiesis* (Maturana and Varela, 1980): the ability of a system to be self-regenerating, or self-forming, such that global level structures and components remain persistent over time, even as lower level components are replaced. Biological systems demonstrate this capacity in many ways, an example being their ability to replenish and rebuild damaged cells in injured tissues.

The dynamics of a complex system are greatly determined by its connectivity, i.e., the arrangement of, and degree of interaction between, its constituent components. Connectivity in ecosystems, which is generally calculated as a function of the interactions between species in the food web, has been the subject of considerable discussion, particularly with regards to the relationship between connectivity and ecosystem stability (Pimm et al., 1991; Pimm, 1984). Conrad (1995), for example, discussed food web connectivity with reference to the stages of succession in ecosystem development. He suggested that in the early stages there is considerable redundancy in the food web, corresponding to system dynamics exhibiting a large density of closely packed stable states. As an ecosystem matures to a climax state, redundant interactions are phased out, and the system dynamics settle into a "basin of attraction that is deep and wide relative to perturbations the community is experiencing" (p. 619). Conrad suggested that this may leave the community increasingly vulnerable to major destabilisation, but that "this is much less likely to be the case if most interactions are weak. Large numbers of weak interactions allow for efficient information processing, high amenability to evolution, effective self-organisation, and buffering against perturbation" (p. 620).

Persistent ecosystems in their climax states may have characteristic degrees of connectivity. In this context, Kauffman (1993) used network models to demonstrate that at a "critical degree of connectivity", an ecosystem will reach a persistent dynamic state "at the edge of chaos", in which organisms have evolved to the highest fitness levels. Similarly, Green (1994) used simple cellular automata models to show that there is a necessary "critical connectivity" that is required in a natural system before an epidemic or forest fire will spread, for example, and stated that this degree of connectivity seems to be characteristic of all living systems. Green also discussed the possibility that the connectivity of an ecosystem may fluctuate with the seasons, becoming poorly connected in a harsh, dry season and then more richly connected during a rainy season.

The concept that an ecosystem may have an organisational structure that keeps it in a non-equilibrium, yet persistent state ("at the edge of chaos", in Kauffman's terms) may be explained by self-organised criticality theory, as promulgated by Bak and colleagues (Bak, 1996; Bak et al., 1988; Bak and Chen, 1991; Gell-Mann, 1994). This theory states that some large, many-body systems "naturally evolve to a critical state in which a minor event starts a chain reaction that can affect any number of elements in the system" (Bak and Chen, 1991, p. 46). It has been used to explain the temporal distributions of a wide range of natural events, such as earthquakes and extinctions (e.g., Adami, 1993), leading to the speculation that a system such as an ecosystem may be as much perturbed by the impact of a large meteorite as by the drop of a pin. An equally important implication is that "composite systems never reach equilibrium but instead evolve from one metastable state to the next" (Bak and Chen, 1991, p. 46). (It is implied here that the system under discussion is open, and subject to continuous inputs.) In paleobiology, this type of dynamics has been termed "punctuated equilibrium" (Gould and Eldridge, 1977), since species have been observed to undergo gradual evolutionary change, punctuated by rapid, radical changes at certain times in the history of Earth. Thus, despite the presence of strong evolutionary and environmental pressures, there are long periods of relatively stable species records. Similarly, the tendency of ecosystems to remain globally stable despite underlying change has been noted by many researchers (Holland, 1995) and is, of course, typical of any autopoietic system.

## 1.2. Bringing together complexity and ecosystem engineering

Persistence of species, food web stability, quasi-equilibrium, etc., are maintained in ecosystems via a number of phenomena, such as autopoiesis, self-organised criticality, and emergence, that are not fully understood. It is known, however, that these phenomena are universally characteristic of ecosystems, and living systems in general. Thus, before a set of principles for the engineering of living systems can be devised, the conditions that give rise to these phenomena need to be elucidated.

If complexity can be defined, in engineering terms, as: "the measure of uncertainty in achieving a design that meets stated functional requirements" (Nam Suh, Complexity in Engineering Conference, November 1999, MIT), then living systems such as ecosystems are perhaps the most complex systems of all. It may be that an ecosystem cannot be precisely designed at all, and must instead be synthesized or evolved, or created with a combination of approaches (Langton, 1989; Bonabeau and Theraulaz, 1994; see Steels (1995; 1994) for a discussion with reference to artificial intelligence). Minimally, recent research in complex system studies would suggest that ecosystem engineering must be performed with an increased awareness of the conditions that give rise to the complex patterns and structures in these systems which, until recently, have been treated in a much simpler manner, or completely overlooked.

A goal of this project, therefore, has been to reproduce some of these complex features in simulation so that the conditions that give rise to their occurrence may be studied. One of the greatest challenges of this goal has been the development of a modelling approach with which ecosystems can be represented as complex systems. As discussed in Chapter 2, recent models developed in the fields of artificial life and computational ecology far surpass traditional single-predator, single-prey models in their ability to capture the essence of complex, evolving systems. These models, however, are not sufficiently configurable, nor are they comprehensive enough in their representation of ecosystems, to be of use as ecosystem engineering tools. The model developed for this project is based on a combination of some of these new approaches, yet is far more detailed and broader in scope that most others of its kind. It is also completely configurable, so that it may be useful for the exploration of different ecosystem designs, etc. In developing the model, rather than focusing on the accurate representation of any one component, the goal has been to include most of the major components of an ecosystem, to at least some degree of detail, in order that the overall behaviour of the system might be explored.

Although the model may be used to represent a wide variety of different ecosystem types, it was configured and tested in the context of the EcoCyborg Project, and has therefore been developed to address some of the issues related to ecocyborg engineering. Again, the importance of complexity is stressed. Certainly, with regards to the development and testing of intelligent control systems for ecosystems, it is probably most important to train the control system to deal with a type of global comportment that is complex; a type of comportment that could somehow come closer to mimicking the kind of variability and change that would be expected to occur in natural ecosystems.

## 1.3. Summary

The body of this thesis has been written as a collection of papers (see preface). In the first of these (Chapter 2), a literature review of the current trends in ecosystem modelling, particularly those methods which have been adopted to model ecosystems as complex systems, is provided. In Chapter 3, an outline of the approach that was used to create the ecosystem model is presented, including a detailed description of the simulation program in which it is encoded. In Chapter 4, the manner in which the ecosystem encompassment is modelled is described, and the major pathways of material cycling in the modelled ecosystem are outlined. The results of overall, long-term baseline simulations are presented, in which it is shown that the model produces reasonable results with trends in material cycling and accumulation that make sense. In Chapter 5, the manner in which plant life is modelled is described in detail and results of simulations are presented that illustrate various aspects of vegetation dynamics. In Chapter 6, the way that animals are modelled is described, and sample simulation results for multi-trophic level ecosystem configurations are presented. Lastly, the thesis is summarized with a discussion of the overall behaviour of the model when implemented in simulation, and the degree to which it succeeds in exhibiting the kind of complex comportment that is typical of living systems.

## 1.4. References

Adami, Chris. 1993. *Self-organized criticality in living systems.* Working paper, W. K. Kellogg Radiation Laboratory, 106-38, California Institute of Technology, Pasadena, California, USA. December 20, 1993.

Assad, A. M. and N. H. Packard. 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. Bourgine, 143-152. Boston: MIT Press.

Bak, P. 1996. *How Nature Works.* New York, NY: Springer-Verlag.

Bak, P. and K. Chen. 1991. Self-organized criticality. *Scientific American* 264(1): 46-53.

Bak, P., C. Tang and K. Wiesenfeld. 1988. Self-organized criticality. *Physical Review A* 38(1): 364-373.

Bar-Yam, Y. 1999. Complexity rising: From human beings to human civilization, a complexity profile. On-line paper: http://www.necsi.org/civilization.html.

Bonabeau, E. W. and G. Theraulaz. 1994. Why do we need artificial life? *Artificial Life* 1(3): 303-325.

Clark, O.G. 1999. *The Characterization of Biosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Conrad, M. 1995. The ecosystem as an existential computer. In *Complex Ecology,* eds. Patten, B.C. and Jørgensen, S.E., 609-622. Englewood Cliffs, NJ: Prentice Hall.

Fahse, L., C. Wissel and V. Grimm. 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.

Gell-Man, M. 1994. *The Quark and the Jaguar*. New York: W.H. Freeman & Co.

Gould, S.J. and N. Eldridge. 1977. Punctuated equilibrium: The tempo and mode of evolution reconsidered. *Paleobiology* 3: 114.

Green, D. G. 1994. Connectivity and complexity in landscapes and ecosystems. *Pacific Conservation Biology* 1(3): 194-200.

Haken, H. 1977. *Synergetics: An Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology.* Berlin; New York: Springer-Verlag.

Holland, J.H. 1995. *Hidden Order: How adaptation breeds complexity.* Addison-Wesley Publishing Co, Reading, MA.

Holling, C.S. 1996. Engineering resilience versus ecological resilience. In *Engineering within Ecological Restraints*, ed. P. Schulze, 31-43. Washington, DC: National Academy Press.

Kauffman, S. 1993. *The Origins of Order: Self-organisation and Selection in Evolution.* New York: Oxford University Press.

Langton, C. 1989. Artificial Life. In *Artificial Life*, ed. C. Langton, 1-47. Redwood City, CA: Addison-Wesley.

Mandelbrot, B. 1982. *The Fractal Geometry of Nature.* San Fransico, CA: W.H. Freeman.

Maturana, H.R. and F. Varela. 1980. *Autopoeisis: The organization of the living.* Dordrecht, Holland: D. Reidel Publishing Co.

Pimm, S.L. 1984. The complexity and stability of ecosystems. *Nature* 307: 321-

326.

Pimm, S.L., J.H. Lawton and J.E. Cohen. 1991. Food web patterns and their consequences. *Nature* 350: 669-674.

Prusinkiewicz, P. 1994. Visual models of morphogenesis. *Artificial Life* 1(1/2), 61-74.

Solomonoff, R.J. 1964. A formal theory of inductive inference. Part I. *Information and Control,* 7, 1-22.

Steels, Luc. 1995. The Homo Cyber Sapiens, the Robot Homonidus Intelligens, and the 'artificial life' approach to artificial intelligence. In *Proc. Burda Symposium on Brain-Computer Interfaces,* Muenchen, February 1995.

Steels, Luc. 1994. The artificial life roots of artificial intelligence. *Artificial Life* 1(1/2), 75-110.

Wolpert, D. and W. Macready. 1997. Self-dissimilarity: An empirically observable complexity measure. In *Proc. International Conference on Complex Systems,* Nashua, NH, 21-26 Sept.

# Chapter 2. Incorporating Complexity in Ecosystem Modelling

## Abstract

This article is a review of how the ecosystem modelling process 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. It is argued that when an ecosystem is conceptualized as a complex system, an *object-based* approach is a very appropriate representational method, and various types of object-based ecosystem models are reviewed. These are divided into three categories: individual-based models, agent-based models, and cellular automata. Several different types of computational approaches and programming platforms that are applicable to object-based ecosystem modelling are then reviewed.

## 2.1. 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, Ashby 1956). 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 small-world network[1] of many components whose aggregate behaviour is both due to, and gives rise to, multi-scale structural and dynamical patterns which are not inferrable from a system description that spans only a narrow window of resolution (adapted from NECSI, 1999). 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

---

[1] A distributed network 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 (Watts and Strogatz, 1998).

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 specialities, ranging from economics and social policy to biology, physics and modern visual arts (Heylighen, 1997; Waldrop, 1992; Santa Fe Institute, 1999).

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 Maturana and Varela (1980): *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.

## 2.2. Modelling an ecosystem as a complex system

The integration of the fields of complex systems studies and ecology has had 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 2.1) 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 2.1, when an ecosystem is analysed and subsequently modelled in the context of complex system studies, all of these stages are affected.

### *2.2.1. 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

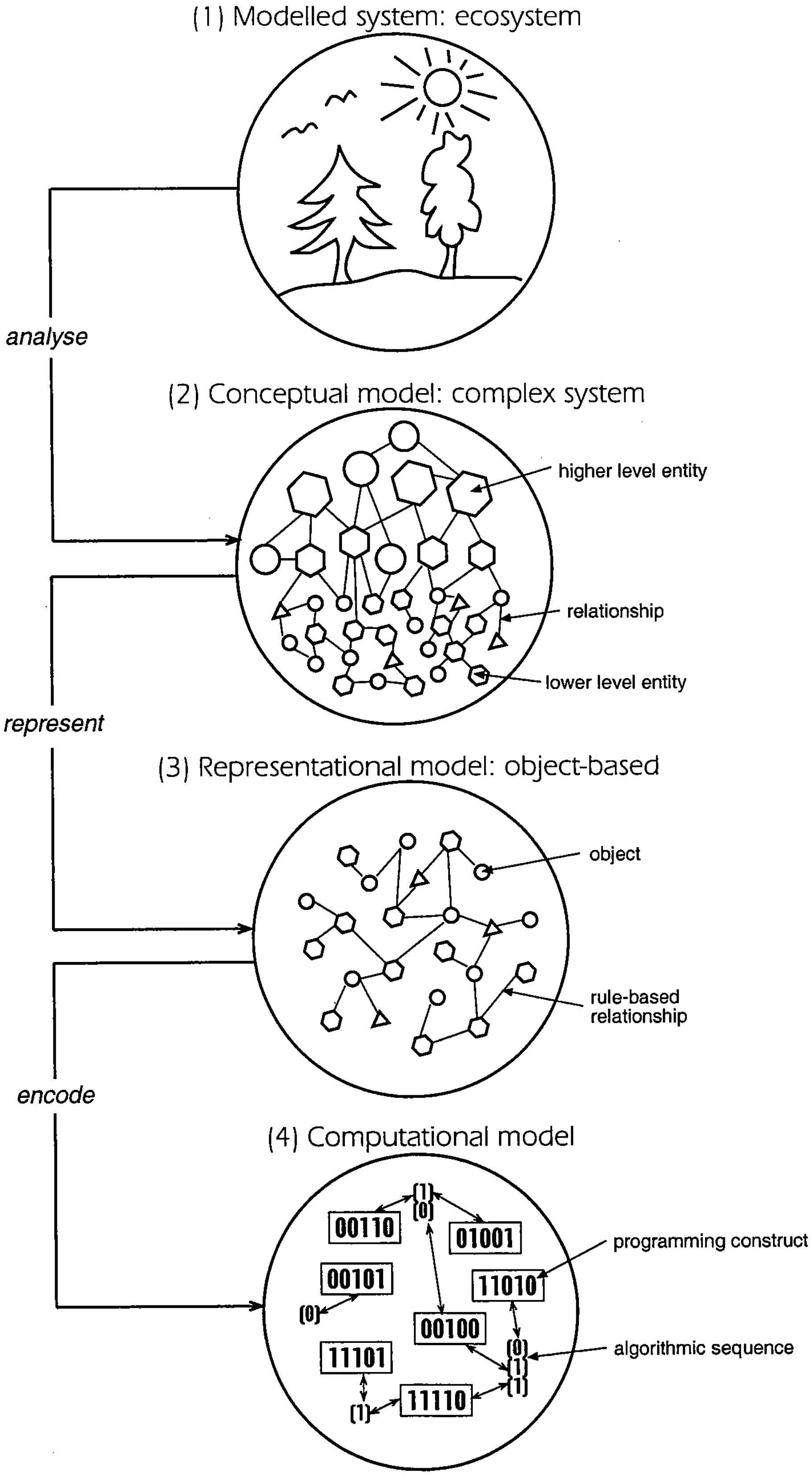


Figure 2.1: The modelling process.

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 (Assad and Packard, 1992), the almost periodic boom-bust cycles in a predator-prey relationship, the flocking of birds (Reynolds, 1987), and the natural tendency of organisms in a population to be distributed log-normally over a terrain (Taylor, 1992).

### *2.2.2. 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 (Kawata and Toqeunaga, 1994).

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 (Gell-Man, 1994).

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 (Millonas, 1994).

### *2.2.3. The computational model*

In order to be used in simulation, an object-based representation must be encoded in computational form (Figure 2.1), 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 2.4, 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.

## 2.3. 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.

### 2.3.1. *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 (Judson, 1994; see also: Grimm, 1999, 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 (Reynolds, 1987) 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 (1994), 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. (1998) 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 (1998) 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 (Botkin et al., 1972; Huston, 1992). 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 (Booth, 1997) has developed one of the only individual-based multi-species 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.

At present, there are no detailed, large-scale, multi-species individual-based ecosystem models. This is partly due to a number of implementation concerns, which have been outlined by McCauley et al. (1993): (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., Palmer, 1992). Also, for many individual-based models, the selection of appropriate attribute values can be difficult and very time-consuming (Hogeweg and Hesper, 1990). 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.

### *2.3.2. 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 (Taylor et al., 1988; Maes, 1994)). 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, 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. In the second, an overriding genetic algorithm is used to explicitly select for certain traits or behaviours.

The modern version of the genetic algorithm is attributed to Holland (1975; 1995) and, in its simplest form, is as follows (Mitchell and Forrest 1994, p. 268):

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 (Mitchell, 2000). 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 (1997) 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" (1992; 1994a; 1994b; 1994c). 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 (O'Callaghan and Conrad, 1992; Rizki and Conrad, 1985) 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 (Bonabeau and Theralauz, 1994). 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 (1989, 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.

### *2.3.3. 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 (1984), 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 (1990) 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 (1988) 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 (Green, 1993; Hogeweg, 1988) 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 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 (Miramontes et al., 1993)). Hogeweg (1988) 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.

## 2.4. 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.

### 2.4.1. *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. (1991) and NeXT Software (1996).

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 (1993), the predator-prey model of Plant and Stone (1991), and the animal foraging model of Beecham and Farnsworth (1998). 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 (1993) 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.

### *2.4.2. 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. (1992) 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 (Swarm Development Group, 1999; Langton et al., 1995-99; Minar et al., 1996; Hiebeler, 1994). 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. Although it was not written specifically for ecosystem modelling, there are presently a number of ecosystem models being developed with SWARM, including a port of "Gecko".

ECOSIM (Lorek and Sonnenschein, 1998; Sonnenschein et al., 1994-99) 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 has been written solely for the purpose of ecosystem modelling, and is probably the most extensive and well documented simulation platform of its kind. 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.

SWARM and ECOSIM are probably the most extensive object-based simulation platforms that have been developed for public use. The recent (i.e., post-1998) edition of SWARM is the only platform known to the authors that has been seriously considered for large-scale ecosystem modelling. Further examples of software available to model and simulate ecosystems in an object-based manner are given in the review by Lorek and Sonnenschein (1999). None of these, 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 (Gathmann and Williams, 1998). 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".

## 2.5. 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 homogeneous spatial conditions, their predictions may very well match those of an equivalent object-based approach (McCauley et al., 1993; Wilson, 1998). In addition, 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, no 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 approach in the development of large-scale ecosystem models, increased computing capacity is rapidly alleviating some of the other drawbacks. The result is that the object-based approach is becoming more feasible, and may be used more commonly for large ecosystem modelling projects in the future. In 1998, for example, researchers from several U.S. groups who are developing models to represent large regions such as the Florida Everglades, Cheasapeake Bay, and the Columbia River watershed, visited the Santa Fe Institute to assess and review the SWARM object-based simulation platform and to "coordinate efforts in dynamic spatial modeling" (SFI, 1998).

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 (Holling, 1993). It is for this reason that object-based modelling approaches are being considered for large-scale environmental assessment projects. The comportment of an ecosystem is fundamentally complex, and cannot be accurately predicted by models that are constrained to a narrow range of variables (Holling, 1996; Jørgensen et al., 1998). 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 comportment 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.

Due to the fundamental unpredictability of an ecosystem's comportment (Gell-

Man, 1994), simulations using object-based computational models cannot be used to precisely forecast events over the long-term. They may, however, provide useful illustrations of the nature of ecosystem dynamics and of the mechanisms that give rise to unexpected events. 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.

## 2.6. References

Ashby, W.R. 1956. *An Introduction to Cybernetics.* London: Chapman & Hall.

Assad, A. M. and N. H. Packard. 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. Bourgine, 143-152. Boston: MIT Press.

Beecham, J.A. and K.D. Farnsworth. 1998. Animal foraging from an individual perspective: an object orientated model. *Ecological Modelling* 113: 141-156.

Booth, G. 1997. Gecko: A continuous 2-D world for ecological modelling. *Artificial Life* 3(3): 147-163.

Botkin, D. B., J. F. Janak and J. R. Wallis. 1972. Some ecological consequences of a computer model of forest growth. *Journal of Ecology* 60: 849-872.

Bonabeau, E. W. and G. Theraulaz. 1994. Why do we need artificial life? *Artificial Life* 1(3): 303-325.

Gathmann, F.O. and D.D. Williams. 1998. Inter-site: a new tool for the simulation of spatially realistic population dynamics. *Ecological Modelling* 113: 125-139.

Gell-Man, M. 1994. *The Quark and the Jaguar.* New York: W.H. Freeman & Co.

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. Amsterdam: IOS Press.

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.

Heylighen, F. 1997. Publications on complex, evolving systems: A citation-based survey. *Complexity* 2(5): 31-36.

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.

Hogeweg, P. 1988. Cellular automata as a paradigm for ecological modeling. *Applied Mathematics and Computing* 27: 81-100.

Hogeweg, P. and B. Hesper. 1990. Individual-oriented modelling in ecology. *Mathematical and Computational Modelling* 13(6): 83-90.

Holland, John H. 1995. *Hidden Order: How adaptation breeds complexity*. Reading, MA: Addison-Wesley Publishing Co.

Holland, John H. 1975. *Adaptation in Natural and Artificial Systems. An introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor, MI: University of Michigan Press.

Holling, C.S. 1996. Engineering resilience versus ecological resilience. In: *Engineering within Ecological Constraints*, ed. P. Schulze, 31-43. Washington, DC: National Academy Press.

Holling, C.S. 1993. Investing in research for sustainability. *Ecological Applications* 3(4): 552-555.

Hraber, P.T. and B.T. Milne. 1997. Community assembly in a model ecosystem. *Ecological Modelling* 103: 267-285.

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. New York: Routledge, Chapman and Hall, Inc.

Jørgensen, S, H. Mejer, and S. Nielsen. 1998. Ecosystem as self-organizing critical systems. *Ecological Modelling* 111: 261-268.

Judson, O. 1994. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution* 9(1): 9-14.

Kawata, M. and Y. Toquenaga. 1994. From artificial individuals to global patterns. *Trends in Ecology and Evolution* 9(11): 417-421.

Langton, C. 1990. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* 42: 12-37.

Langton, C. 1989. Artificial Life. In *Artificial Life*, ed. C. Langton, 1-47. Redwood City, CA: Addison-Wesley.

Langton, C., N. Minar, R. Burkhart, M. Askenazi and G. Ropella. 1995-99. The Swarm Simulation System. Web-based documentation: *http://www.santafe.edu/projects/swarm*, Santa Fe Institute, Santa Fe.

Letcher, B. H., H. A. Priddy, J. R. Walters, and L. B. Crowder. 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.

Lorek, H. and M. Sonnenschein. 1998. Object-oriented support for modelling and simulation of individual-oriented ecological models. *Ecological Modelling* 108: 77-96.

Lorek, H. and M. Sonnenschein. 1999. Modelling and simulation software to support individual-based ecological modelling. *Ecological Modelling* 115: 199-216.

Maes, P. 1994. Modeling adaptive autonomous agents. *Artificial Life* 1(1/2): 135-162.

Maley, C. C. and H. Caswell. 1993. Implementing i-state configuration models for population dynamics: an object-oriented programming approach. *Ecological*

*Modelling* 68: 75-89.

Maturana, H.R. and F. Varela. 1980. *Autopoeisis: The organization of the living.* Dordrecht, Holland: D. Reidel Publishing Co.

McCauley, Edward, William G. Wilson and Andre M. de Roos. 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.

Miramontes, O., R. Solé and B. Goodwin. 1993. Collective behaviour of random-activated mobile cellular automata. *Physica D* 63: 145-160.

Millonas, M.M. 1994. Swarms, Phase Transitions, and Collective Intelligence. In *Artificial Life III*, ed. C. Langton, 417-445. Reading, MA: Addison-Wesley.

Minar, N., R. Burkhart, C. Langton and M. Askenazi. 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.

Mitchell, M. 2000. *Life and evolution in computers.* Working paper, Biophysics Group, Los Alamos National Laboratory.

Mitchell, M. and S. Forrest. 1994. Genetic Algorithms and Artificial Life. *Artificial Life* 1(3): 267-289.

Murdoch, W. W., E. McCauley, R. M. Nisbet, W. S. C. Gurney and A. M. de Roos. 1992. Individual-based models: Combining testability and generality. In *Individual-Based Models and Approaches in Ecology*, eds. D. DeAngelis and L. Gross, 18-35. New York: Routledge, Chapman and Hall, Inc..

NECSI. 1999. New England Complex Systems Institute Web Site. http://www.nesci.org. NECSI, New Hampshire.

NeXT Software Inc. 1996. *Object Oriented Programming and the Objective-C*

*Language*. Next Software Inc.

O'Callaghan, J. and M. Conrad. 1992. Symbiotic interactions in the EVOLVE III ecosystem model. *Biosystems* 26: 199-209.

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. New York: Routledge, Chapman and Hall, Inc..

Plant, R. and N. Stone. 1991. Object-oriented models. In *Knowledge-based Systems in Agriculture,* ed. R. Plant, 259-285. New York: McGraw Hill.

Ray, T. S. 1994a. TIERRA.DOC: Documentation for the Tierra Simulator V4.1.

Ray, T. S. 1994b. An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life* 1(1/2): 179-209.

Ray, T. S. 1994c. Evolution, complexity, entropy and artificial reality. *Physica D* 75: 239-263.

Ray, T. S. 1992. Evolution, ecology and optimization of digital organisms. Santa Fe Institute working paper 92-08-042.

Reynolds, C. W. 1987. Flocks, herds, and schools: A distributed behavioral model (Proceedings of SIGGRAPH '87). *Computer Graphics* 21(4): 25-34.

Rizki, M. M. and M. Conrad. 1985. EVOLVE III: a discrete events model of an evolutionary ecosystem. *Biosystems* 18: 121-133.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen. 1991. *Object-Oriented Modeling and Design.* New Jersey: Prentice Hall.

Santa Fe Institute. 1998. Annual Research Report. Santa Fe Institute, New Mexico.

Santa Fe Institute. 1999. SFI Bulletin 14(2). New Mexico: Santa Fe Institute.

Silvert, W. 1993. Object-oriented ecosystem modelling. *Ecological Modelling* 68: 91-118.

Sonnenschein, M., F. Köster, H. Lorek and U. Vogel. 1994-1999. ECOSIM Web-based documentation. *http://offis.OFFIS.Uni-Oldenburg.DE/projekte/ecotools/ecosimintro.htm*

Swarm Development Group. 1999. The Swarm Simulation System. Web-based documentation: *http://www.swarm.org*

Taylor, C., D. Jefferson, S. Turner and S. Goldman. 1988. RAM: Artificial life for the exploration of complex biological systems. In *Artificial Life,* ed. C. Langton, 276-295. Reading, MA: Addison-Wesley.

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. New York: Routledge, Chapman and Hall, Inc.

Waldrop, M. 1992. *Complexity.* New York: Touchstone Books.

Watts, D. and S. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393: 440-442.

Wilson, W. G. 1998. Resolving discrepancies between deterministic population models and individual-based simulations. *The American Naturalist* 151(2): 116-134.

Wolff, W. F. 1994. An individual-oriented model of a wading bird nesting colony. *Ecological Modelling* 72: 75-114.

Wolfram, S. 1984. Cellular automata as models of complexity. *Nature* 311: 419-426.

## *Connecting text between Chapters 2 and 3*

In Chapter 2, a review of the current state of ecosystem modelling was presented, in which the term, *object-based* was introduced to describe the class of models in which components of a system are represented as discrete entities that interact according to rule-based relationships. Three different types of object-based models were highlighted, including: individual-based models, agent-based models, and cellular automata. Different examples of these that have been reported in the literature were discussed. Next, object-oriented programming was introduced, and several examples were given of object-oriented simulation platforms that have been developed to assist in the creation of object-based ecosystem models. All of this was placed within the context of a "modelling process", in which first a conceptual model of a system is created, next a representational model is designed, and lastly the representational model is encoded into a computational model that can be used in simulation.

Chapter 3 is an introduction to the overall modelling approach and programming techniques that have been adopted to represent and encode a highly configurable ecosystem model. The ecosystem is represented with an object-based approach that contains elements of both individual-based and cellular automata types of models. Despite the availability of several excellent object-based simulation platforms[1], the model, and associated simulation routines, have been encoded entirely from scratch, in ANSI compliant C. This was done in order to ensure that

---

[1] At the beginning of this project, a beta version of the Swarm Simulation System was tested and evaluated by L. Parrott. At that time, it was not sufficiently well developed (or bug free!) to serve as a platform in which the EcoCyborg ecosystem model could be written. The current version of Swarm runs under two supported systems (Solaris machines running Sun SPARC and Intel x86 machines running GNU/Linux), as well as many unsupported systems that support GNU C development. Although it might be interesting to port the ecosystem model to Swarm 2.0.1, some of the advantages of our approach would be sacrificed, including code portability, ability to integrate the model with other components written in different programming langauges (e.g., future control system modules), and the coding flexibility provided by using a mixed OOP/procedural approach.

the model could be compiled to run under any operating system and hardware configuration, both at the present time and in the foreseeable future. As described in Chapter 3, the computational model, although written in a procedural language, contains some elements of an object-oriented program. This provided a great deal of coding flexibility: although not bound by the programming structure of an OOP language, object-oriented ideas were borrowed when it was convenient to do so, and strictly procedural programming techniques were used for other routines for which this approach was most efficient (e.g., weather).

This work is a unique contribution to the domain of object-based ecosystem modelling, since it is of considerably greater scope than most other object-based models. Due to its extensive representation of all of the major aspects of an ecosystem's structure and function, and due to the large numbers of objects included in a simulation, this model is currently the most comprehensive object-based ecosystem model of which the authors are aware.

The routines described in Chapter 3 are encoded in the files "ecowea.c" (main simulation kernel routines), "encompas.c", "control.c" and "bio.c", which are available for viewing on the enclosed CD-ROM (Appendix B).

Chapter 3 was coauthored by L. Parrott and R. Kok and is being prepared to be submitted for publication.

# Chapter 3. Implementation of a Generally Configurable Object-based Ecosystem Model

## Abstract

In this article, key features of an object-based ecosystem model that has been developed for use as an engineering research tool are described. A brief summary of the model is given, followed by a detailed explanation of the computational approach and programming techniques used to implement it in simulation. The model is in many ways a unique contribution to the fields of both ecological modelling and ecosystem engineering: First, its scope (i.e., the number of different types of ecosystem components that are included) is much broader than that of most other ecosystem models, and key processes are represented at relatively high spatial and temporal resolutions (10 metres and 10 minutes, respectively). Second, the model is entirely object-based, wherein every abiotic and biotic component in the system is represented as a distinct entity. Thus, each organism, or small group of organisms, is treated as an individual object that lives in a spatially explicit environment composed of cells arranged in a 2-D lattice. Third, the model is completely configurable, such that a wide range of ecosystem configurations and their corresponding initial conditions can be specified for simulation. Thus, both the biological composition (i.e., number and type of species, initial population sizes, etc.) and the environment (i.e., terrain and atmosphere) of an ecosystem can be specified. Lastly, the computational approach used to encode the model is a unique blend of procedural and object-oriented techniques that have been developed specifically to treat such a large, object-based model. The model will be implemented in simulation to study a broad range of ecosystem design and management issues.

## 3.1. Introduction

The current trend in ecosystem modelling has been to move away from 'top-down' approaches in which global cycles and population dynamics are described with several overriding equations, towards the adoption of object-based, 'bottom-up' approaches (Judson, 1994; Grimm 1999). In an object-based model, all of the components in a system at a given level of resolution are represented by discrete objects whose functions and interrelationships are described by simple rules. Thus, with this approach, system level comportment is elicited in simulation through the aggregation of events occurring at the object level. This 'bottom-up' synthesis of global level structures and dynamics enables the study of mechanisms that give rise to certain complex features of a system, such as self-organisation or emergence, that are not easily studied with 'top-down' approaches (Kawata and Toquenaga, 1994). For this reason, the object-based approach is becoming an increasingly popular method by which to represent living systems, the structure and dynamics of which have always been difficult to represent with analytical models. The three most common types of object-based models used in ecology are: individual-based models, in which objects represent individual organisms; agent-based models, in which objects represent abstract organisms that have the capacity to evolve; and cellular automata, in which objects most often represent small spatial patches in a landscape (Parrott and Kok, 2000a).

Although there are many advantages to object-based modelling, there are two major disadvantages. The first of these is that most object-based models require the specification and input of a large number of very specific parameter values (McCauley et al., 1993; Hogeweg and Hesper, 1990). In an individual-based ecosystem model for example, in which there are several different species types represented, data corresponding to the attributes of individual organisms for each species is required. With the exception of the small number of species that have been studied intensively due to their economic importance or scientific interest, few, or none, of the necessary field data are available. The second major disadvantage is computational intensity. When object-based models are used in simulation, their performance obviously decreases with increasing numbers of objects, thereby limiting the overall size of the model that can be accommodated within the computational capacity of the available hardware. Due to these disadvantages, most object-based ecosystem models developed to date have been of rather limited scope, with a focus on higher-resolution population dynamics and food webs, as opposed to global biospheric processes. Such models are generally of two types: they are either very specific representations of the behaviour of individuals belonging to a few, well known species, or they are abstract, general models that represent generic organisms. The latter are most commonly used to explore principles of theoretical ecology.

The scope (i.e., extent of treatment, or range of component types included) as well as the resolution (i.e., spatial or temporal scale) of an ecosystem model determines, to some degree, the approach that is used in its development. As discussed above, most high resolution ecosystem models tend to be of narrow scope, and are usually written in an object-based manner. In contrast, with models of a broader scope, usually only ecosystem components discernible at a fairly low resolution are represented. This type of model is often composed of interrelated sub-models which are based on sets of simultaneous algebraic or differential equations. Although, theoretically, models developed with this approach can represent a system at any resolution, usually the biological components are represented at the population or landscape level, and biospheric processes corresponding to mass and energy flows are described on a fairly coarse scale. Examples of such models are: Hybrid v.3.0 (Friend et al., 1997) which was written to represent the terrestrial biosphere within a larger whole-Earth system model, and the modular General Ecosystem Model (GEM) which was developed as a research tool to simulate a variety of ecosystem types (Fitz et al., 1996). Due to the extensive spatial area represented by these models, in neither are animal life forms included, since they were developed with a focus on biospheric processes rather than higher-resolution population dynamics.

In this article, a unique ecosystem model is described that is of a broad scope, and yet is object-based. It contains representations of atmospheric, terrestrial and biological components. The model is primarily individual-based, in that most organisms are represented as discrete objects, called "instances". However, to make the system computationally feasible, some instances represent small lumps of individuals. Up to 100,000 biological instances can be accommodated in the model, each of which may belong to one of a possible 1000 species of plants and animals. All of the instances are situated on, and move about upon, a spatially explicit landscape which is modelled as a grid of rectangular cells. As with most object-based models, the model described here, with its current level of resolution, is exceedingly computationally intensive, despite some of the efforts made to reduce the load by using lumped biological objects. Simulations do, however, proceed at a reasonable speed on currently available desktop computers.

The ecosystem model is entirely configurable, allowing for the complete specification of an ecosystem starting state and initial conditions. To accomplish this, a number of configuration programs have been written to facilitate the model specification procedure and also to reduce some of the time required with regards to the input, and selection, of appropriate parameter values. The configuration programs draw upon a database of pre-defined species types that enable the automatic generation of reasonably coherent and realistic biological instances. The set of configuration programs, together with the simulation program itself, form a flexible engineering research tool that can be used to explore the relationships between external forcing functions, and the constitution (structure and composition) and resultant comportment (dynamic behaviour) of complex ecological systems.

A detailed summary of the model's scope, the programs written to enable its configuration, and the manner in which it is encoded in computational form, is presented here.

## 3.2. Background

The model has been developed as part of a larger research initiative, the EcoCyborg Project, the long-term goal of which is to learn how to engineer highly complex, autopoietic systems, or *biosystems* (Maturana and Varela, 1980; Clark & Kok, 1999). This project was established to address basic questions regarding the design, construction, maintenance, repair, and control of all kinds of autopoietic systems with the aim of developing a general theory, or set of principles, of biosystems engineering. At present, the focus is on *ecocyborgs*: ecosystems that have been augmented with sufficiently sophisticated control mechanisms so as to render them substantially autonomous (Parrott et al., 1999). Understanding how to engineer ecocyborgs has a wide range of possible applications in areas such as space-based life support, intensive agriculture, environmental remediation, and natural resource management.

The particular case study that is being investigated in detail is that of a hypothetical ecocyborg installed in an isolated space station (see Figure 3.1). This setting was selected since it is convenient to model (due to the well defined system boundary), and because of its relevance to the future establishment of off-Earth human settlements. The station is envisioned as a materially closed, cylindrical structure containing an artificially constructed ecosystem together with supporting infrastructure, including a sophisticated control system and a large material storage chamber. The current short term goal of the project is to create a detailed computational model of the space station ecocyborg, and to use this in simulation to explore how to engineer such a system. Thus, the model described here has been developed in partial pursuit of this goal, and has been configured to represent the ecosystem component of the hypothetical ecocyborg. A weather model and a rudimentary control system have also been developed, and are linked with the ecosystem model during simulation runs. A more sophisticated control system is currently also under development.

### 3.3. Objectives

The objective of the work reported here was, therefore, to create a generally configurable computational model that could be used to simulate the behaviour of ecosystems. Although it was tested and configured to represent the ecocyborg ecosystem, the goal was not to represent any specific terrestrial ecosystem, nor even any particular type of ecosystem. Rather, the intent was to create a virtual system that, when appropriately configured, would exhibit at

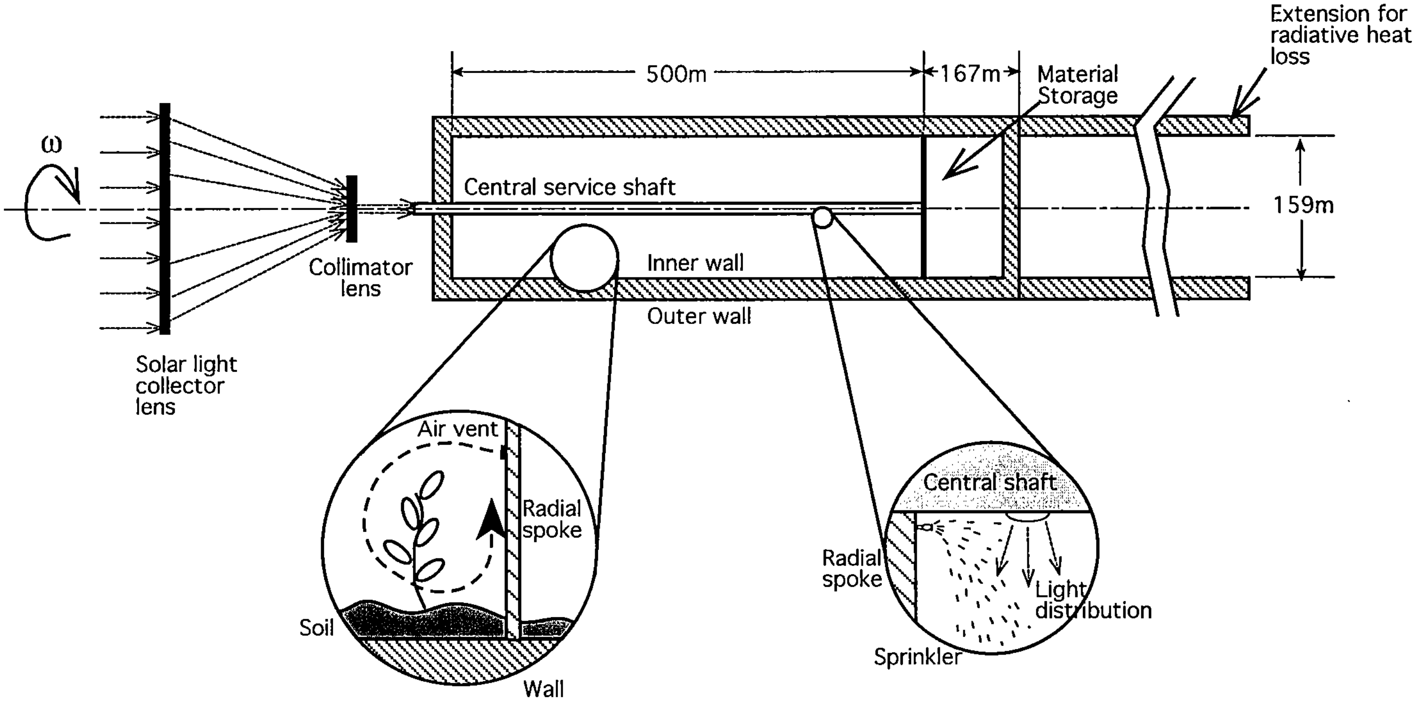


Figure 3.1: The space station ecocyborg (not to scale).

least some of the features that are common to all large ecological assemblages, and whose behaviour would reflect, to some degree, the general class of comportment demonstrated by complex biological networks in response to naturally occurring forcing functions. Also, since this model may be linked in the future with an intelligent control system so as to explore the engineering of ecocyborgs, an additional requirement was that it reproduce sufficiently complex behaviour in order to present a challenge to the controller's cognitive components. Thus, development of the model proceeded according to the following set of criteria:

With regards to scope and resolution,

- the model should represent all major abiotic and biotic components in an ecosystem

- relationships between these components should include spatial, environmental and trophic types of interactions

- components should be represented in a fairly high-resolution, object-based manner

With regards to the computational model and associated simulation program,

- it should be possible to specify different configurations and initial states, and this should be done in a time-efficient manner, e.g., by drawing upon a database of pre-defined object descriptions

- the program code should be hardware independent, run on various platforms without adjustment, and have a long useful life

As will become evident throughout the rest of the article, these objectives and criteria underlie many of the decisions that were made regarding the creation of the ecosystem model and its implementation in simulation.

### 3.4. General Approach

The model is based on a conceptual approach in which an ecosystem is regarded as a highly complex biological network composed of a large number of inter-related components. This network is then represented in an object-based manner and encoded in computational form. A differentiation is made here between the "representational model" which is an abstract, human language (symbolic, mathematical or linguistic) description of a system, and the "computational model" which is the actual, encoded version of the representational model that can be compiled and implemented in simulation. The representational model, therefore, is an object-based model of the ecosystem, in which the state of each object is described by a number of state variables corresponding to properties, and the object's functionality is described by discrete mathematical or rule-based expressions. The computational model consists of abstract data structures that contain the states of objects in a computer's memory space, and algorithmic sequences in which the object functions are encoded. The computational modelling code, as well as a simulation kernel that consists of routines to handle time management, data collection and storage, program initialisation and other associated tasks, form a "simulation program". A number of configuration programs are used to specify parameter values for the simulation program.

Development of the various configuration programs and the simulation program was done on personal computers, primarily under IBM's OS/2 Warp 4 operating system. The configuration programs were written in, and compiled with, Microsoft Basic 7 for OS/2; the simulation program was written in ANSI compliant C so as to ensure cross-platform compatibility. When the model is configured to have 100,000 biological components, together with a complete encompassment (see below for descriptions of these), the simulation program requires between 150-200 Mb of memory during its execution. The simulation is time-driven, and current experiments are being run with an increment of 10 (simulated) minutes per cycle. To date, the program has been compiled and run on a number of platforms, including OS/2 Warp, Macintosh OS 8.5 and Windows 95/98. Compilation for Windows 95/98 and Macintosh OS/8.5 was done with Metrowerks CodeWarrior Pro v.4; compilation for OS/2 was done with IBM's Visual Age C++ development tools. Execution speed on Pentium III-based desktop computers is approximately 6 simulated years per real day when there are 25,000 extant plant instances; with 25,000 animal instances, execution slows to less than 1 simulated month per real day. In the following sections, a brief description of the representational model is provided, followed by a detailed explanation of the structure and implementation of the simulation program, including the steps required to configure a simulation starting state.

## 3.5. The Representational Model

To facilitate the representation of systems such as the space-station EcoCyborg, an overriding boundary condition of the model is the assumption that the modelled ecosystem is situated inside a materially closed cylindrical shell (the size of which is configurable via parameter values input to the model). Additionally, the ecosystem is considered to be divided into three main parts, referred to here as realms. These are: the biological component, encompassment, and material storage realms. The biological component realm consists of all the living organisms in the ecosystem (plants and animals). The encompassment is the abiotic environment in which the biological components exist, consisting of an atmosphere and a terrain (soil and water). Mass is regularly cycled through the biological components, terrain, and atmosphere, and is also transferred (by a control system) between these and the storage realm, in order to control a number of key variables. The storage realm is a reserve of materials that has been included in imitation of the Earth's biosphere, which is naturally buffered by large masses of water and other materials (e.g., the oceans or the lithosphere). The overall material composition of the three realms is shown in Figure 3.2. Every part of the system is assumed to be composed of one or more of nine basic compounds that are made up of five different elements, the total masses of which are accounted during a simulation. The dynamics of the three realms are driven by forcing functions imposed by an external Weather Generator which supplies values for the current temperature (°C), radiation intensity ($W \cdot m^{-2}$) and rainfall rate ($m \cdot s^{-1}$). A control system may or may not be present; in current configurations, some minimal control mechanisms are included.

The objects in the biological component realm are all species "instances", where an instance may be an individual organism, or a small group of organisms, depending on the scale and fecundity of the species. Interactions between instances, and between instances and their environment, form the core of activity within the ecosystem model. Although all of the species have names that suggest some resemblance to physical species of the same name, this resemblance is only partially realised in the model: while none of the instances could be said to be fair representations of their physical species's counterparts, they do, nevertheless, tend to fill the same niche in the virtual system as the corresponding physical ones do in their natural settings. There are, for example, primary producers and consumers whose general behaviour parallels that of natural organisms. As described below, the representation of the biological component realm borrows some concepts from the field of object-oriented programming; thus, with reference to object-oriented terminology, the properties of biological component instances are called "attributes", and their functionality is described with "methods".

The encompassment and material storage realms are also represented as collections of components. The terrain is modelled as a grid of cells, each of

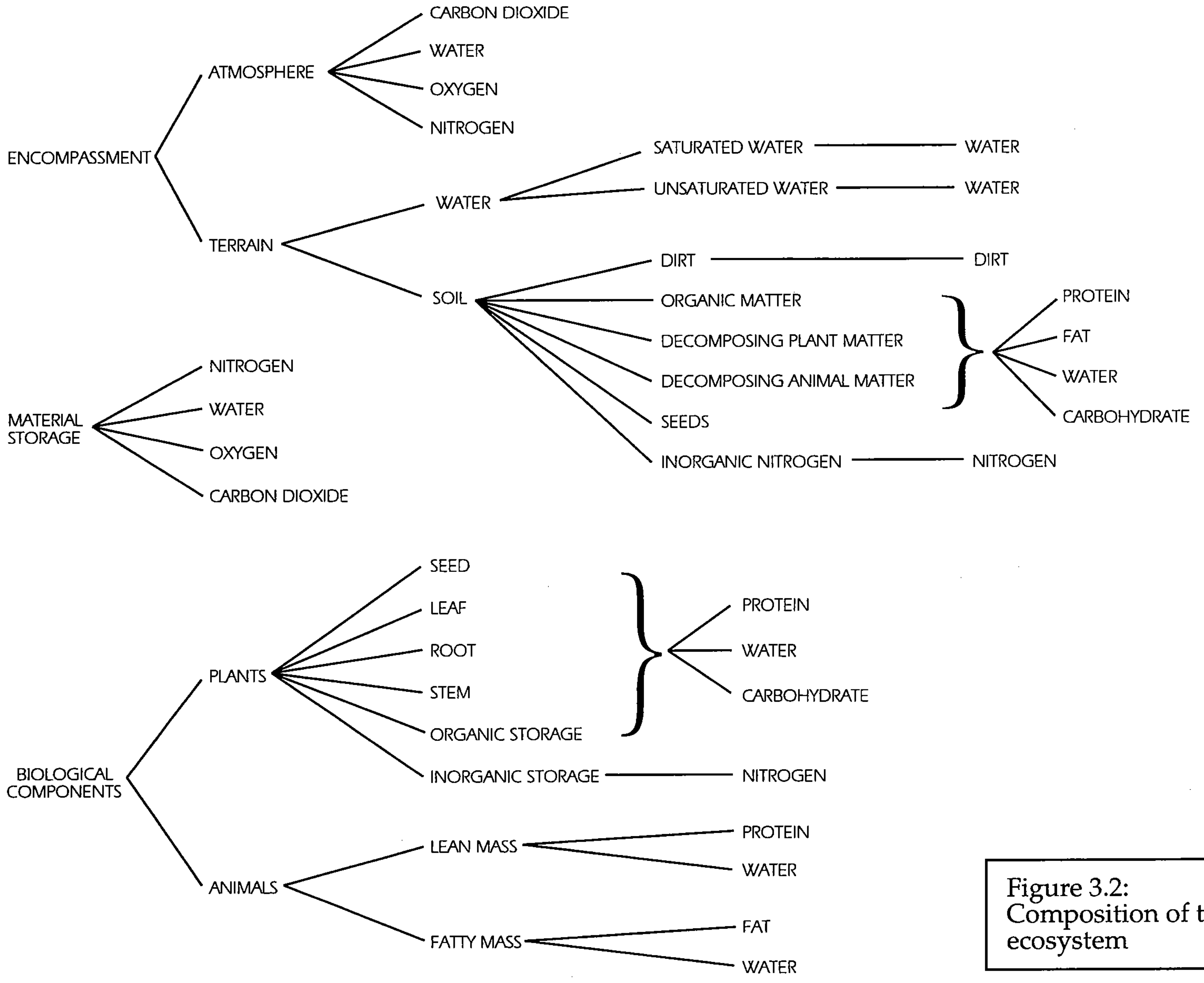


Figure 3.2: Composition of the ecosystem

which has properties whose values are stored in matrices and that can be modified as the result of time driven processes such as decomposition, as well as by single events caused by the actions of organisms. Each grid cell is composed of soil (dirt, inorganic nitrogen, seeds, organic matter, decomposing plant matter and decomposing animal matter) and water (saturated and unsaturated). The atmosphere is modelled as a single component with properties related to its physical make-up (i.e., oxygen, carbon dioxide, water and nitrogen content). The material storage realm is modelled as four chambers, each of which has one property, namely the mass of the compound that is stored inside.

## 3.6. Configuration of the computational model

Since the intent was to create a set of experimental tools with which very different hypothetical ecosystems could be specified and then implemented in simulation, model configurability was a key consideration during its development. Accordingly, a set of programs was written with which the model can be configured to represent any kind of ecosystem, thereby setting its starting state in preparation for simulation. This is done in two stages: *class definition,* during which all the different types, or species, of biological component objects are defined, and *component specification,* in which the compositions and initial states of all three realms are set. During both stages, data files are created that are later read into memory by the simulation program during its initialisation phase. The definition and specification stages are described below. (For further details, see Parrott, 1995.)

### *3.6.1. Definition of biological component classes*

Although the model was not encoded in an object-oriented language per se, some ideas stemming from the object-oriented programming approach were used to facilitate the representation of the biological component realm. In particular, the object-oriented concept of hierarchical classes proved to be a practical way to define and describe the different types of species included in the model. Accordingly, a five-level hierarchy of biological component classes was

- **Autotroph**
    - *Photosynthesizing organism*
        - Tree
            - Needle-leaved
            - Broad-leaved evergreen
            - Evergreen-sclerophyll
            - Broad-leaved deciduous
            - Thornyc
            - Rosette tree
            - Bamboo
        - Climbing plant
            - Woody vine
            - Non-woody vine
        - Shrub
            - Needle-leaved
            - Broad-leaved evergreen
            - Evergreen-sclerophyll
            - Broad-leaved deciduous
            - Thorny
            - Rosette shrub
            - Stem succulent
            - Semishrub
            - Dwarf shrub
        - Epiphyte
        - Herb
            - Fern
            - Grass
            - Sedge
            - Other graminoid
            - Forb
        - Thallophyte
            - Lichen
            - Moss
            - Liverwort

**Figure 3.3a:** Autotroph class hierarchy (species classes omitted)
Key: **Level 1**; *Level 2*; Level 3; Level 4.

- **Heterotroph**
  - *Carnivore*
    - Mammal
      - Ranging mammal
      - Burrower
      - Tree inhabitant
      - House builder
    - Reptile
      - Turtle
      - Lizard
      - Snake
      - Crocodile
    - Amphibian
      - Frog
      - Toad
      - Newt
      - Salamander
    - Bird
      - Raptor
      - Insectivorous bird
    - Arachnid
      - Spider
      - Harvestmen
      - Tick
      - Mite
      - Scorpion
    - Insect
      - Flying insect
      - Crawling insect
  - *Herbivore*
    - Mammal
      - Ranging mammal
      - Burrower
      - Tree inhabitant
      - House builder
    - Reptile
      - Turtle
      - Lizard
      - Snake
      - Crocodile
    - Amphibian
      - Frog
      - Toad
      - Newt
      - Salamander
    - Bird
    - Arachnid
      - Spider
    - Mollusc
      - Snail
      - Slug
    - Insect
      - Flying insect
      - Crawling insect
  - *Omnivore*
    - Mammal
      - Ranging mammal
      - Burrower
      - Tree inhabitant
      - House builder
    - Reptile
      - Turtle
      - Lizard
      - Snake
      - Crocodile
    - Amphibian
      - Frog
      - Toad
      - Newt
      - Salamander
    - Bird
    - Insect
      - Flying insect
      - Crawling insect
  - *Detrivore*
    - Insect
      - Flying insect
      - Crawling insect
    - Worm
      - Earthworn
    - Fungi
      - Mushroom
      - Mould
      - Rust
  - *Parasite*
    - Fungi
      - Mushroom
      - Mould
      - Rust

**Figure 3.3b:** Heterotroph class hierarchy (species classes omitted) Key: **Level 1**; *Level 2*; Level 3; Level 4.

developed, starting with general, abstract classes and working downwards towards specific "species" classes. The first four levels of this hierarchy are shown in Figure 3.3. The two root classes in the hierarchy are differentiated based on trophism, with autotrophs (producers) separated from heterotrophs (consumers of all sorts). Associated with each class are: attributes, corresponding to data values, and methods, corresponding to code. All objects are considered to be instances of species classes that inherit attributes (and the values thereof) and methods from ancestral classes.

To facilitate the creation and storage of biological component class definitions, a set of user interface programs were written. The various programs and data files used for class definition are shown in Figure 3.4. Execution of these programs results in the establishment of a hierarchy of classes (such as that depicted in Figure 3.3). As explained below, when a class is defined, the values of a number of parameters are entered for each associated attribute. These parameters describe the attribute's properties such as its data type and distribution. Class instances are created in the biological components specification stage (see next section) using these parameter values.

The first step in the biological components definition stage is to create lists of all possible attributes (object traits, or state variables) and methods (functions, or abilities) that might be associated with any class in the hierarchy. This is done with the Method and Attribute Definition Programs. Definition of a method simply involves the assignment of a descriptive name and a unique method ID number. The definition of an attribute, however, is slightly more detailed. When an attribute is created, in addition to it being given a name and ID number, its data type (integer, floating point or character) must be specified, as well as its "distribution type", i.e., the manner in which values for this attribute will be distributed when they are generated for objects that have this attribute. Possible distribution types include: binary, normal, log normal and uniform. The distribution type determines how many parameter values will have to be entered for that attribute when it is added to a class definition. The values of

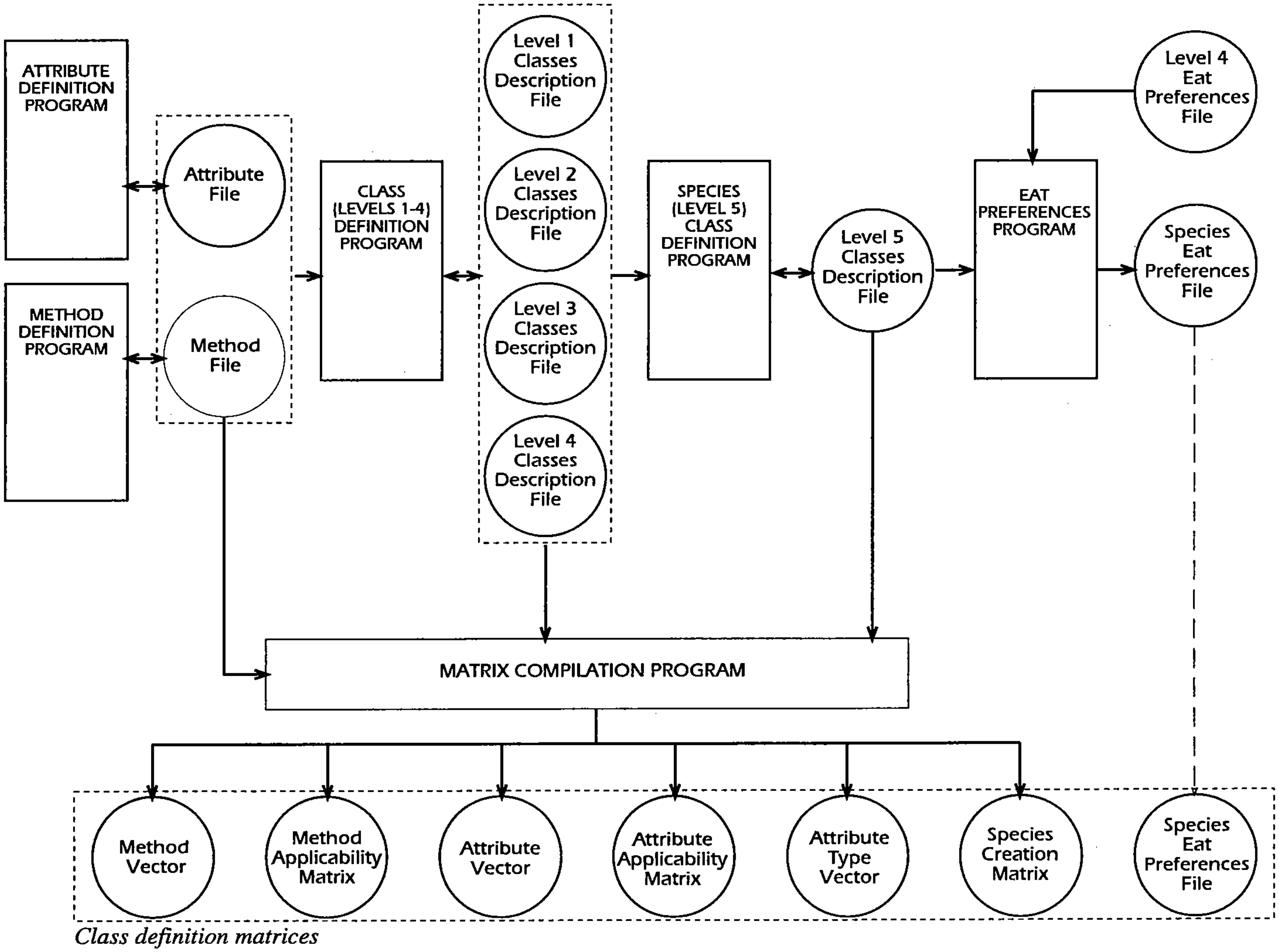


Figure 3.4: Biological component class definition procedure

these parameters (called "attribute creation values") will then later determine the distribution of the related instances's attribute values in a population. Each attribute definition also contains a descriptor tag that specifies whether the attribute's creation values are quite different for "mature" and "juvenile" instances. For example, for the mass and age attributes, these are different, but for gender, they are not. In cases for which there is a difference, there are two parallel sets of attribute creation values that relate to the attribute's distribution in a class definition. This was done so that both mature and juvenile instances can be specified at the start of a simulation; only juveniles are created during a simulation.

After the method and attribute lists have been created, abstract "container" classes (i.e., ones that will have no instances in the computational model) from levels 1 through 4 in the class hierarchy are defined with the Class Definition Program (Figure 3.4). The creation of a class involves the specification of a name and the selection of applicable attributes and methods. Whenever an attribute is added to a class description, the class definition program collects appropriate parameter values. When, for example, the attribute "mass" (for which there are different mature and juvenile creation values), of data type "float" and distribution type "normal" is added to a class, the program will prompt for two sets of creation values, corresponding to the mean, standard deviation, minimum and maximum values for that class. Thus, with this approach, it is quite possible that two classes can be created with identical sets of associated attributes and methods, but be quite dissimilar because of the differences between their attribute creation values. Lastly, species classes (level 5) are created with the Species Class Definition Program. First, a species class definition is compiled from the definitions of four ancestor classes (a level 5 class must have one ancestor in each of the higher levels). Next, additional species-level attributes and methods may be added. Also, the creation values of inherited attributes may be modified. After all species classes have been fully defined, the non-interactive Matrix Compilation Program is used to re-order all of the biological component definitions into a form suitable for the biological component specification stage.

The generation of appropriate initial attribute values for new instances presented a number of issues that had to be considered as the definition programs were being written. First, in order to allow for variation amongst instances of the same species, it was decided that they should not have identical sets of attribute values at the start of their lives, but that these values should be distributed in some way. This is why various distribution types can be specified for each attribute. Thus, as instances of a given species are created, the initial values of their attributes are generated in accordance with the distribution functions specified. The second issue that had to be considered was that instances are created under two different circumstances. The more common situation is creation during execution of the simulation. Such instances are all juveniles, which might be birthed, hatched or germinated. The more complicated situation occurs at time zero, i.e., at the beginning of a simulation, since at this time, the ecosystem must be initialized with mature and juvenile instances for each of the species (e.g., fully grown trees as well as seeds; baby mammals as well as mature ones). At this point, fully functional and coherent instances must be created, without having access to, or simulating, their past (although an incubation scheme is being considered for future work, in lieu of the method described here). This issue was addressed, as described above, by collecting two sets of creation values for some age-sensitive attributes: one for juvenile instances and one for mature instances. Both sets are used to generate initial populations for system startup, but only the first set is made available to the simulation program since, during its execution, no instances other than juveniles are created. Also, a number of attributes are treated as "special cases" during instance creation. These attributes require detailed treatment to ensure the coherence of instances, especially of the mature instances created by the specification programs. Examples of special cases include attributes whose values are related, such as gender and the state of being pregnant. Lastly, the values of a number of attributes are dependent upon the time of year (e.g., the mass of leaves on a deciduous tree). To ease model development, a decision was made to always begin a simulation in the spring season. Thus, for time sensitive attributes, all of the creation values correspond to a species's usual springtime state.

All the species in the ecosystem are interrelated by means of a multi-trophic level food web which is defined with the Eat Preferences Program (Figure 3.4). The output from this program is an "eat preferences" matrix in which each consumer species is assigned an "eat preference" value for every other species in the system. These values are all between 0 and 1, and represent the likelihood that a consumer would eat an instance from the target species, given a selection of food choices of equal availability (cannibalism was excluded; all elements on the diagonal are zero). To simplify the eat preferences definition procedure, a matrix of seed values, based on level 4 ancestry, is supplied to the Eat Preference Program, and these values are then assigned at the species level and are used as initial default values. In this way, all herbivorous burrower species might, for example, be assigned an eat preference value of 0.85 for all grass species. The preference values can be modified on a case-by-case basis to enable more species-specific feeding habits. Thus, if one burrower species rarely consumes any but one or two species of grass, the eat preference values for the others would be set much lower.

The representation of biological components in an object-oriented manner proved to be both conceptually appealing and practical, having many of the advantages that are usually claimed for object-oriented models. The approach facilitates model configuration, readily allowing for the definition of new species (many of which share the same superclasses), as well as the modification of those already defined. In this regard, a definition procedure such as the one used here is advantageous for the development of large ecosystem models for which many related species types need to be defined. Further development and refinement of the object code is also accommodated: the class descriptions created during the definition stage only contain names of methods; since each method is implemented in the computational model as a C-function, the contents of the methods themselves can easily be replaced or changed without requiring the modification of a class definition. For example, all species that inherit from the photosynthesizing organism class have the method "photosynthesize" as part of their class description. Presently, in the C-function that corresponds to the photosynthesis method, the process is represented with a fairly simple series of expressions. If, however, it is deemed necessary in the future to use a different set of expressions, this can be accomplished through the re-writing of a single routine.

### *3.6.2. Specification of an ecosystem and its starting state*

Configuration of the composition and corresponding starting state of an ecosystem for a particular simulation is done during the specification stage. The steps in this procedure are illustrated in Figure 3.5. As described below, each realm of the ecosystem is configured in turn.

#### 3.6.2.1. Specification of the encompassment and material storage realms

The size, structure and composition of the abiotic realms of an ecosystem are almost completely configurable (i.e., except for the overall governing decisions, such as their being a terrain, an atmosphere, etc.). The size of the ecosystem's enclosure, or shell, is the first item that needs to be specified. The required dimensions are input to the Enclosure Specification Program (Figure 3.5) with which the area and volume of the encompassment and material storage realms is then computed. Next, a terrain topography is generated with the Terrain Specification Program (Clark et al., 1997) (the terrain component of the encompassment covers the curving interior wall of the cylindrical enclosure). This software can be used to model a wide variety of landscapes; it outputs a matrix of elevations on a grid of any desired resolution (corresponding to the degree of spatial resolution of the ecosystem model). The Encompassment and Storage Specification Program is then used to place appropriate materials in the encompassment and storage chambers. With this program, a number of system properties and initial conditions can be specified, including: the initial masses of materials in each of the storage chambers, the atmospheric composition and

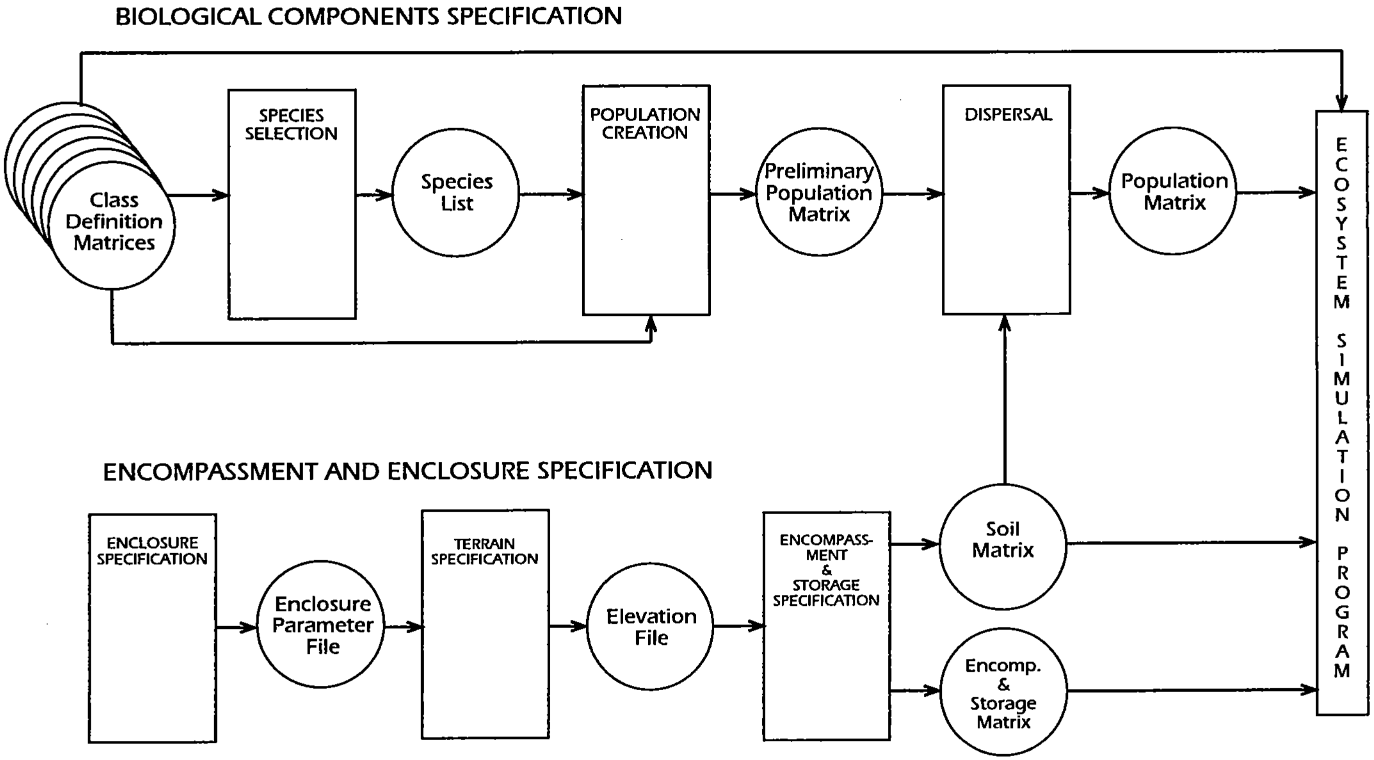


Figure 3.5: Ecosystem specification procedure.

pressure, and certain soil properties such as initial organic matter content, particle density, porosity, and initial degree of saturation with water. A terrain specification file (the Soil Matrix) containing the mass of each substance (saturated water, unsaturated water, decomposing matter, dirt, etc.) in each grid cell is then generated. Other specified initial state values are also written to file.

#### 3.6.2.2. Specification of the biological components

The populations of founding instances with which the ecosystem model is initialized are created with a set of three biological component specification programs (Figure 3.5). The first of these, the Species Selection Program, is used to select the species to be included in the ecosystem model from the list of all species classes previously defined. The chosen species list may include all, or only a few, of those available. Next, the Population Creation Program is executed. With this program, the number of instances to create for each selected species is specified and each instance is subsequently generated with a unique set of attribute values, randomly chosen in accordance with the distribution types and creation values. As discussed above, in order to avoid starting the ecosystem with only juvenile instances, for each species, both mature and juvenile instances are created. Lastly, the Dispersal Program is used to assign each instance an initial location on the terrain.

It should be noted that, although instances of varied degrees of maturity are created during the ecosystem specification stage, this does not ensure that the ecosystem as a whole will be in, or near, a climax state. Thus, although the specification of mature instances is done in an attempt to mitigate some of the initial transient behaviour in a modelled ecosystem, it is understood that even with this approach, it is highly unlikely that a fully developed and persistent initial system state would be specified.

## 3.7. Description of the simulation program

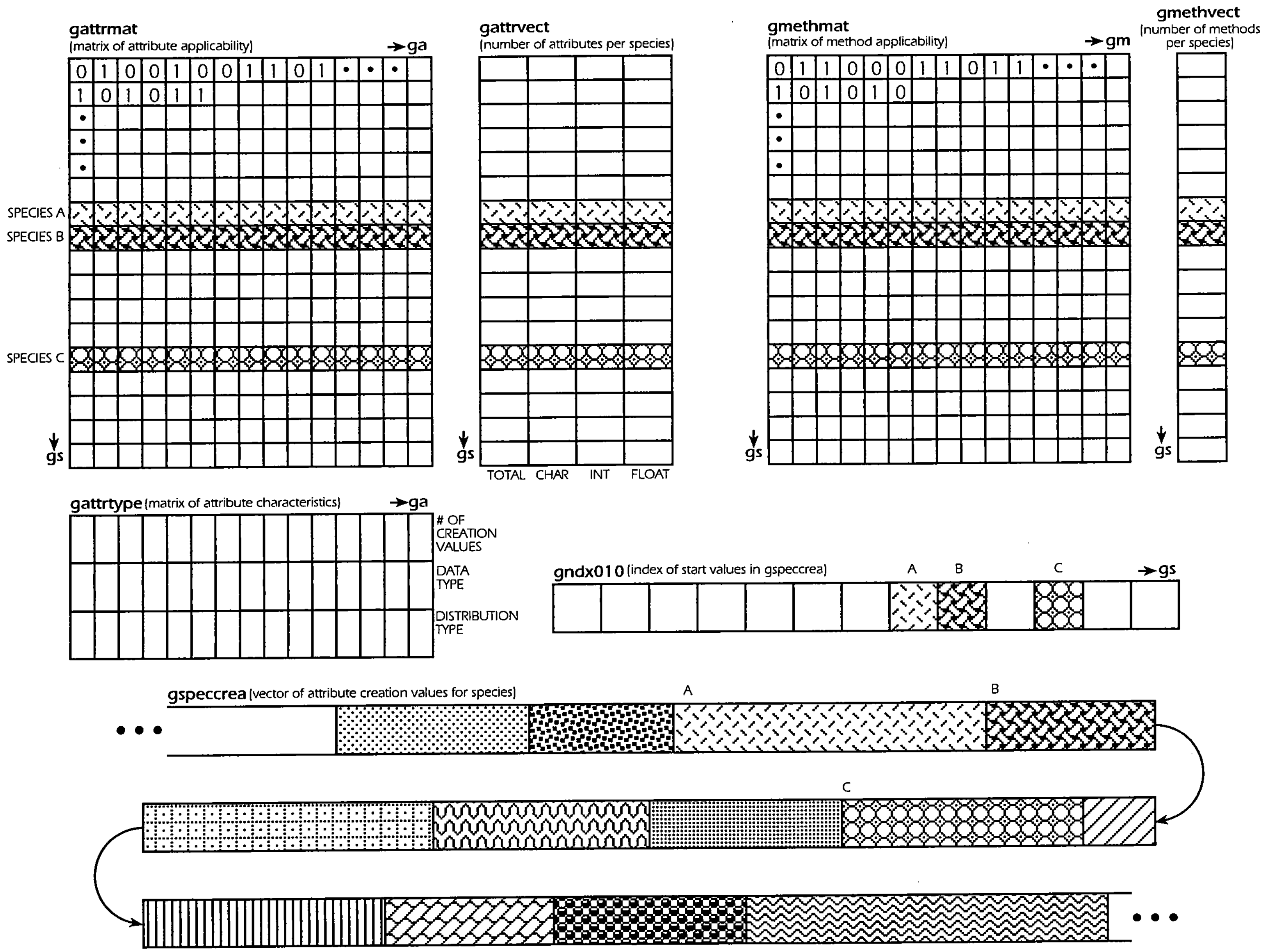
The general approach that has been adopted with regards to implementation of the model in simulation is to store the state of every object in large, globally accessible arrays, and to then continuously update these states (using methods and other computational routines) as the program is executed. A description of the major data structures used in the program is given below, followed by an overview of the program's execution sequence.

### *3.7.1. Storage and retrieval of biological component class definitions*

In this model, biological components have no "genetic memory"; new instances do not inherit attributes from their progenitors but are instead created according to their species's norms, as described above. Therefore, all species class definitions are retained in memory during a simulation, and are used to generate appropriate initial attribute values whenever a new instance is created (birthed, hatched, germinated, etc.). The manner in which the class definitions are stored is illustrated schematically in Figure 3.6. The dimensions of the matrices correspond to the total number of species classes (gs), the total number of attributes (ga), and the total number of methods (gm). All of these matrices contain information collected during the definition and specification stages described above, except for one of them (gndx010), which contains indexing information. Class definitions do not change over the course of a simulation, thus the contents of these matrices are fixed.

The species class definitions are stored in seven major matrices (Figure 3.6). The "attributes applicability matrix" (gattrmat) contains one row per species and one column per attribute. In each species row, for each attribute that is applicable to that class, a 1 is present in the corresponding column; all other locations contain zeros. The applicability of methods is stored similarly in the matrix gmethmat. The total number of applicable attributes for each species class, along with the total number of attributes of each data type (integer, floating point or character) are stored in the "attributes vector" (gattrvect). Likewise, the total number of applicable methods for each species class is stored in the "methods vector" (gmethvect). The attribute properties (data type, distribution type, number of creation values) reside in the "attribute type matrix" (gattrtype). Although there

Figure 3.6 (*facing page*): Data structures used by the simulation program for the storage and retrieval of biological component class definitions.


gattrmat
(matrix of attribute applicability)
→ga
0 1 0 0 1 0 0 1 1 0 1 • • •
1 0 1 0 1 1
•
•
•
SPECIES A
SPECIES B
SPECIES C
gs
gattrvect
(number of attributes per species)
gs
TOTAL
CHAR
INT
FLOAT
gmethmat
(matrix of method applicability)
→gm
0 1 1 0 0 0 1 1 0 1 1 • • •
1 0 1 0 1 0
•
•
•
gs
gmethvect
(number of methods per species)
gs
gattrtype (matrix of attribute characteristics)
→ga
# OF CREATION VALUES
DATA TYPE
DISTRIBUTION TYPE
gndx010 (index of start values in gspeccrea)
A
B
C
→gs
gspeccrea (vector of attribute creation values for species)
A
B
C
• • •
• • •

is a slight overlap in information content between some of these matrices, this arrangement led to relatively fast recall of this information during program execution. To save memory space, all of the attribute creation values for each species are stored in vector form (gspeccrea), with the start location for each species being available in the index vector, gndx010. Thus, when the creation values for a specific attribute are needed, their start location must be computed as an offset from the species start location, based on the number of attributes prior to that, and the number of creation values stored for each.

The process of generating attribute values for a new instance is as follows: for each applicable attribute, as indicated in gattrmat[species ID][attribute ID], the attribute properties (data type, distribution type, and number of creation values) are determined from gattrtype[attribute ID][1,2 & 3]. The attribute's creation values for the species in question are then gathered from gspeccrea[gndx010[species ID] + .. ] and are sent as arguments to the appropriate attribute generation routine which then returns an attribute value. This value is stored in the new instance's memory space (see below), and the program continues with the next attribute, until all the attribute values have been computed. As in the population creation program of the ecosystem model specification stage, there are a handful of exceptional attributes whose values depend upon the values of others, or upon the time of year, and the values for these are generated by special routines. The methods matrix and methods vector are not actively referenced during instance creation, but are accessed regularly during a simulation as a means of verifying that only applicable methods are called by an instance of a species.

### *3.7.2. Storage and retrieval of biological component class instances*

While a simulation proceeds, as the biological instances interact with one another and with the encompassment, the values of their attributes are constantly being modified. This means that during every simulation interval, the data in the matrices in which the states of instances are stored are updated many times. Thus, it is imperative that an instance's attribute values be easily and quickly accessible. In addition, since physical memory space on the host hardware is generally at a premium, the attribute values for all of the instances must be tightly packed. The approach used to store instance states must, therefore, be conservative in memory usage, as well as flexible. This is especially so since instances from different species have unequal numbers of attributes and therefore require somewhat different amounts of storage space. To meet all of these requirements simultaneously, an appropriate scheme for the storage and retrieval of instances was developed. The major data structures used in this scheme are shown in Figure 3.7.

The current state of every biological instance in the ecosystem is stored in three population vectors that contain the values of its attributes: gpopchar, gpopint, and gpopfloat (the names refer to the vector's respective data types). The start locations of all current biological instances in the system (i.e., where their sets of attribute values start in the population vectors) are stored in the index matrix, gndx001. (Note that, if required, the end location of an instance can be computed from the values in gattrvect, in which the number of attributes of each data type is stored for every species.) When the value of a specific attribute of an instance is needed, its location can then be computed as an offset from the instance start location. When a new instance is created, its set of attribute values is placed at the end of the population vectors. When an instance dies, its space is marked as "garbage space" and a reference to this space is placed in the garbage indices: gndx002, gndx003 and gndx004. Once sufficient garbage is accumulated to warrant activity, the population vectors are re-packed (and all the index vectors are updated accordingly).

As well as the main and garbage indices, a number of meta-indices are maintained, to speed up the processing and retrieval of biological instances under specific circumstances. Although these contain information that is also present elsewhere (either implicitly or explicitly), their utility was judged sufficient to justify the memory space allocated to them, and the overhead activity required to maintain them. Firstly, there is the matrix, gndx011. It is simply a table of the

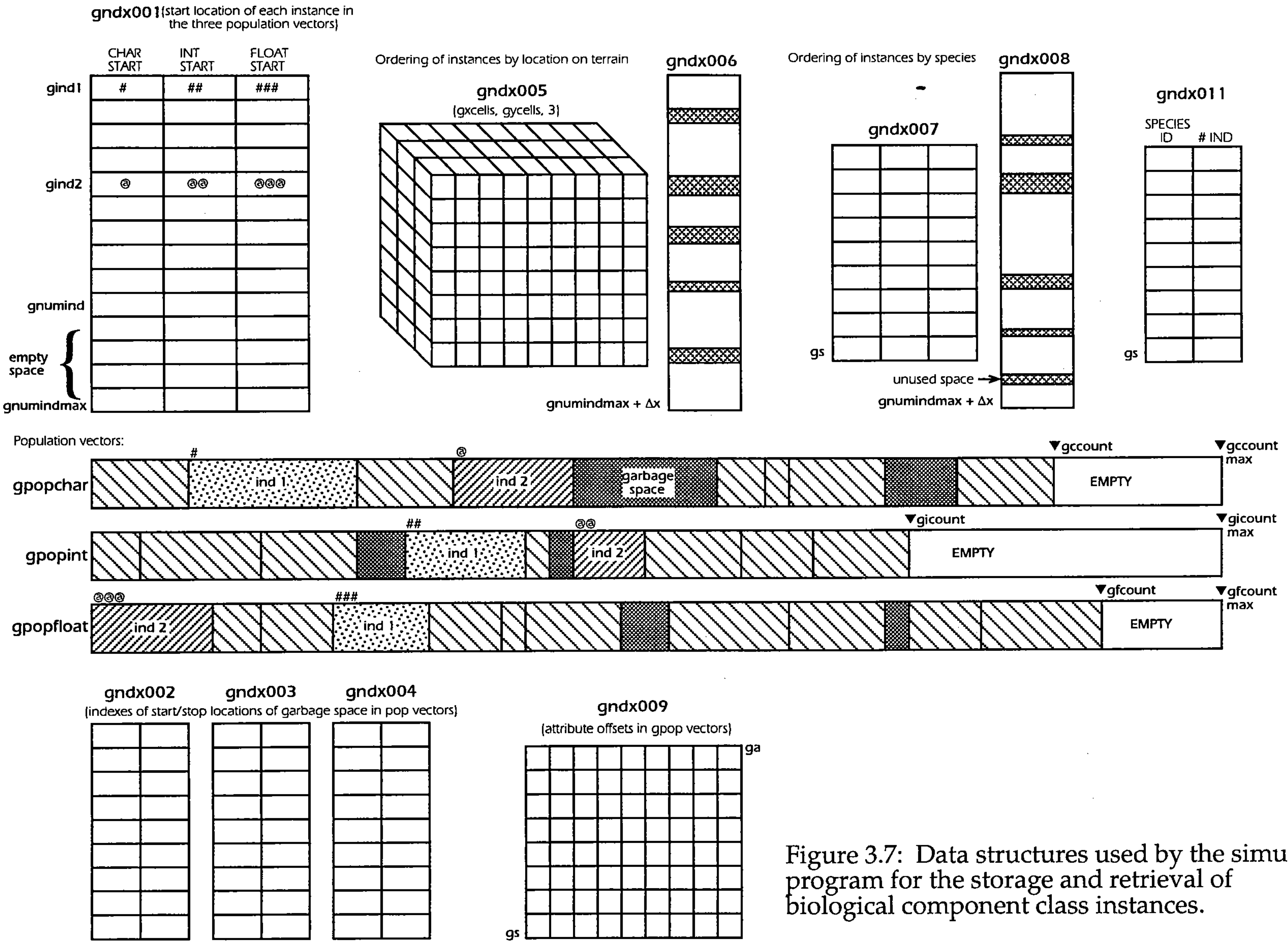


Figure 3.7: Data structures used by the simulation program for the storage and retrieval of biological component class instances.

number of instances of each species that are present. Secondly, the indices, gndx005 and gndx006, are used together to keep track of which instances are situated on each terrain grid cell. In the vector gndx006, the locations of instances in the matrix gndx001 (i.e., the row numbers in that matrix) are grouped together according to terrain grid cell, i.e., for each grid cell there is maintained a list of instances whose references are their row numbers in gndx001. The 3-dimensional matrix gndx005 is a meta-index to gndx006. For each of the terrain grid cells, gndx005 contains the start/stop locations of the list of instances in the gndx006 vector, as well as the end of the total space allocation for that grid cell in the vector. By providing for empty space, the number of times that the vector needs to be re-packed and/or stretched is kept to a minimum. Thirdly, the meta-indices, gndx007 and gndx008, are used to quickly find instances according to species. Like gndx006, gndx008 is a vector of gndx001 row numbers, but grouped according to species membership and, like gndx005, the gndx007 matrix is an index to the start/stop locations for the species groups in the gndx008 vector. The latter also contains a certain amount of empty space to accommodate system change. Fourthly, the offsets from an instance's start locations to each of its attribute values are made available in the gndx009 matrix. These are calculated directly from the information in gattrvect (Figure 3.7).

During a simulation, at any one moment, there can be, at most, two "active" instances: a primary active instance and a secondary active instance whose state is being affected by the primary one[1] (in the simulation, the variables gind1 and gind2 contain the gndx001 row numbers of the two active instances). When required by a method, the values of the active instances's attributes are retrieved from the population vectors by use of the values of their start locations in gndx001, and the attribute offsets in gndx009.

---

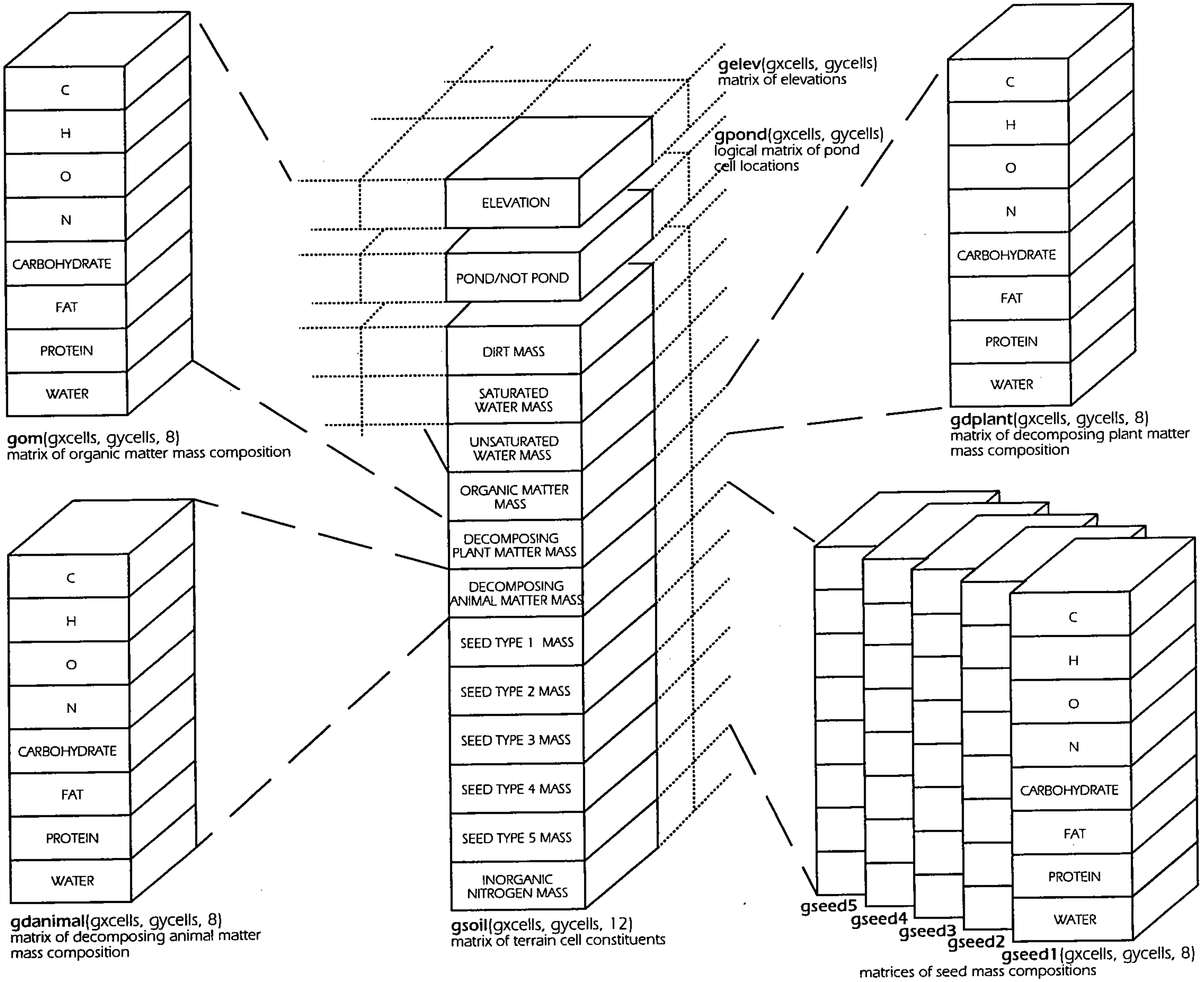
[1] This is an arbitrary choice: in the program, no explicit limit is placed on the total number of active individuals; for more complicated behavioural methods in future versions, it may be interesting to increase this number.

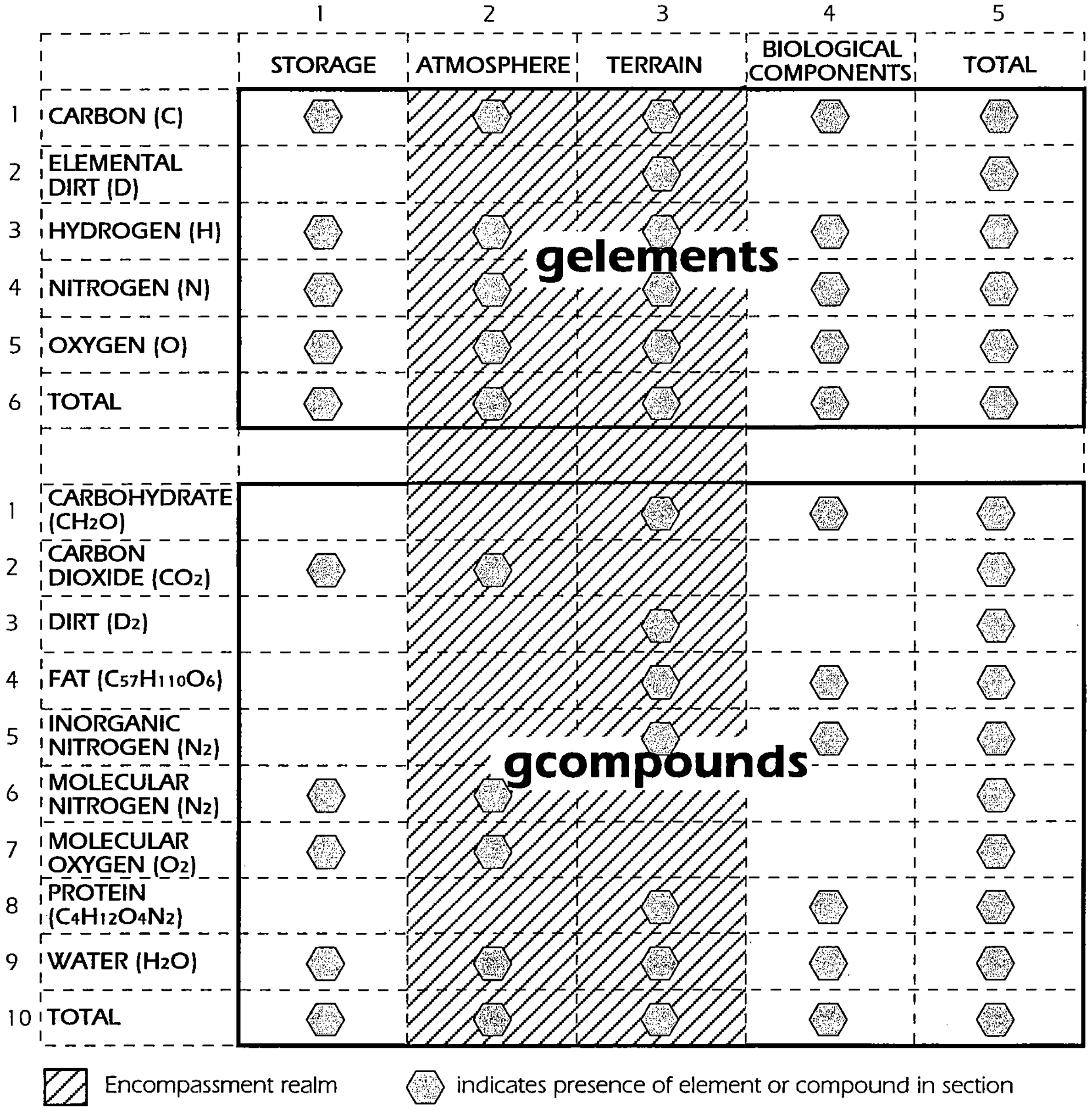
### *3.7.3. Storage and retrieval of encompassment and material storage states*

Whereas in the biological components realm the number of instances of species classes varies continuously, in the encompassment and material storage realms, the numbers of components that are treated as distinct objects are fixed. Thus, the terrain is modelled as a regular, rectangular grid of cell objects, each of which has a number of properties, the values of which describe its state (e.g., elevation, masses of soil and water, etc.). During a simulation, the grid layout and terrain topography (cell elevations) are fixed; the values of all other grid cell properties can be changed, and are updated regularly on a cell by cell basis. The values of most grid cell properties are stored in one main, three-dimensional matrix (gsoil)(Figure 3.8). The elevation of each grid cell is, however, stored in gelev. The gpond matrix is used to keep track of which cells have a water table level that is higher than the soil elevation. In addition, a number of ancillary matrices (gom, gdanimal, gdplant, gseed1, etc.) serve to store more detailed information about the make-up of the organic and decomposing matter fractions of the soil, and of the various groups of seeds. The values of certain global properties, such as soil porosity and hydraulic conductivity are considered to be uniform over the entire terrain, and are not stored separately for each cell. During program execution, the value of the properties of cells are accessed and updated via messages to the encompassment to modify the state of one of its terrain cells by adding or removing material of a certain type. The values of the corresponding array elements are then modified directly to affect these instructions.

The atmosphere, which is assumed to be perfectly mixed, is treated as a single object. The material storage realm is treated as four objects, which represent the four storage chambers. The values of the properties of the atmosphere and material storage objects are stored in two meta-arrays: gelements and gcompounds (Figure 3.9). Equivalent information (i.e., overall compositional values) for the other major parts of the ecosystem is also stored in these arrays. Those values, however, are the results of summation over many thousands of objects. A description of mass and encompassment processes such as water flow

Figure 3.8 (*facing page*): Data structures used by the simulation program to store and update the states of grid cell objects.


C
H
O
N
CARBOHYDRATE
FAT
PROTEIN
WATER
gom(gxcells, gycells, 8)
matrix of organic matter mass composition
gelev(gxcells, gycells)
matrix of elevations
gpond(gxcells, gycells)
logical matrix of pond cell locations
ELEVATION
POND/NOT POND
DIRT MASS
SATURATED WATER MASS
UNSATURATED WATER MASS
ORGANIC MATTER MASS
DECOMPOSING PLANT MATTER MASS
DECOMPOSING ANIMAL MATTER MASS
SEED TYPE 1 MASS
SEED TYPE 2 MASS
SEED TYPE 3 MASS
SEED TYPE 4 MASS
SEED TYPE 5 MASS
INORGANIC NITROGEN MASS
gsoil(gxcells, gycells, 12)
matrix of terrain cell constituents
C
H
O
N
CARBOHYDRATE
FAT
PROTEIN
WATER
gdplant(gxcells, gycells, 8)
matrix of decomposing plant matter mass composition
C
H
O
N
CARBOHYDRATE
FAT
PROTEIN
WATER
gdanimal(gxcells, gycells, 8)
matrix of decomposing animal matter mass composition
C
H
O
N
CARBOHYDRATE
FAT
PROTEIN
WATER
gseed5
gseed4
gseed3
gseed2
gseed1(gxcells, gycells, 8)
matrices of seed mass compositions



gelements

| | | 1<br>STORAGE | 2<br>ATMOSPHERE | 3<br>TERRAIN | 4<br>BIOLOGICAL COMPONENTS | 5<br>TOTAL |
|---|---|---|---|---|---|---|
| 1 | CARBON (C) | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |
| 2 | ELEMENTAL DIRT (D) | | | ⬢ | | ⬢ |
| 3 | HYDROGEN (H) | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |
| 4 | NITROGEN (N) | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |
| 5 | OXYGEN (O) | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |
| 6 | TOTAL | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |

gcompounds

| | | 1<br>STORAGE | 2<br>ATMOSPHERE | 3<br>TERRAIN | 4<br>BIOLOGICAL COMPONENTS | 5<br>TOTAL |
|---|---|---|---|---|---|---|
| 1 | CARBOHYDRATE ($CH_2O$) | | | ⬢ | ⬢ | ⬢ |
| 2 | CARBON DIOXIDE ($CO_2$) | ⬢ | ⬢ | | | ⬢ |
| 3 | DIRT ($D_2$) | | | ⬢ | | ⬢ |
| 4 | FAT ($C_{57}H_{110}O_6$) | | | ⬢ | ⬢ | ⬢ |
| 5 | INORGANIC NITROGEN ($N_2$) | | | ⬢ | ⬢ | ⬢ |
| 6 | MOLECULAR NITROGEN ($N_2$) | ⬢ | ⬢ | | | ⬢ |
| 7 | MOLECULAR OXYGEN ($O_2$) | ⬢ | ⬢ | | | ⬢ |
| 8 | PROTEIN ($C_4H_{12}O_4N_2$) | | | ⬢ | ⬢ | ⬢ |
| 9 | WATER ($H_2O$) | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |
| 10 | TOTAL | ⬢ | ⬢ | ⬢ | ⬢ | ⬢ |

▨ Encompassment realm ⬢ indicates presence of element or compound in section

Figure 3.9: Material composition of the ecosystem as tabulated by the mass accounting matrices, gelements and gcompounds.

and decomposition can be found in Parrott and Kok, 2000b.

### *3.7.4. Major procedural loops and method calling*

The simulation program consists of an initialization phase and an iteration phase. (An overall flowchart is presented in Figure 3.10.) During the initialization phase, the system configuration is read in from the specification files, and all of the program's global variables are set based on the input parameter values. (The names of *global* variables are valid everywhere in the program, as opposed to *local* variables which exist only within certain routines. A list of global variables is given in the article's Appendix.) Global variables include all of the items discussed above related to class definitions, food web relationships, and the states of objects, as well as various other model parameters corresponding to the values of physical constants. The values of simulation kernel parameters, such as the size of the time increment, the number of time steps to be executed, and the frequency with which to save the system state, are also read in from file during the program initialization phase. After initialization, the program proceeds to the iteration section. For each cycle, the time is incremented and a number of routines are called and executed. First, forcing function values for the time increment are obtained from the Weather Generator. Next, the BiologicalComponentLoop, EncompassmentLoop and ControlLoop routines are called in turn. It should be noted that, since the material storage realm is only modified by the encompassment and control routines, there is no "MaterialStorageLoop" as such. Flowcharts of the three major routines are shown in Figure 3.11, Figure 3.12, and Figure 3.13.; these are described further in the paragraphs below.

In the BiologicalComponentsLoop routine, each instance is dealt with in turn and instructed to "Act". Each Act method contains a list of activities (in the form of rule-based instructions that call other methods) that an organism might perform during any given time increment. In general, all species that inherit from the same level 4 ancestor share the same Act method, so that instances of related species all perform similar sequences of activities. When an Act method has been

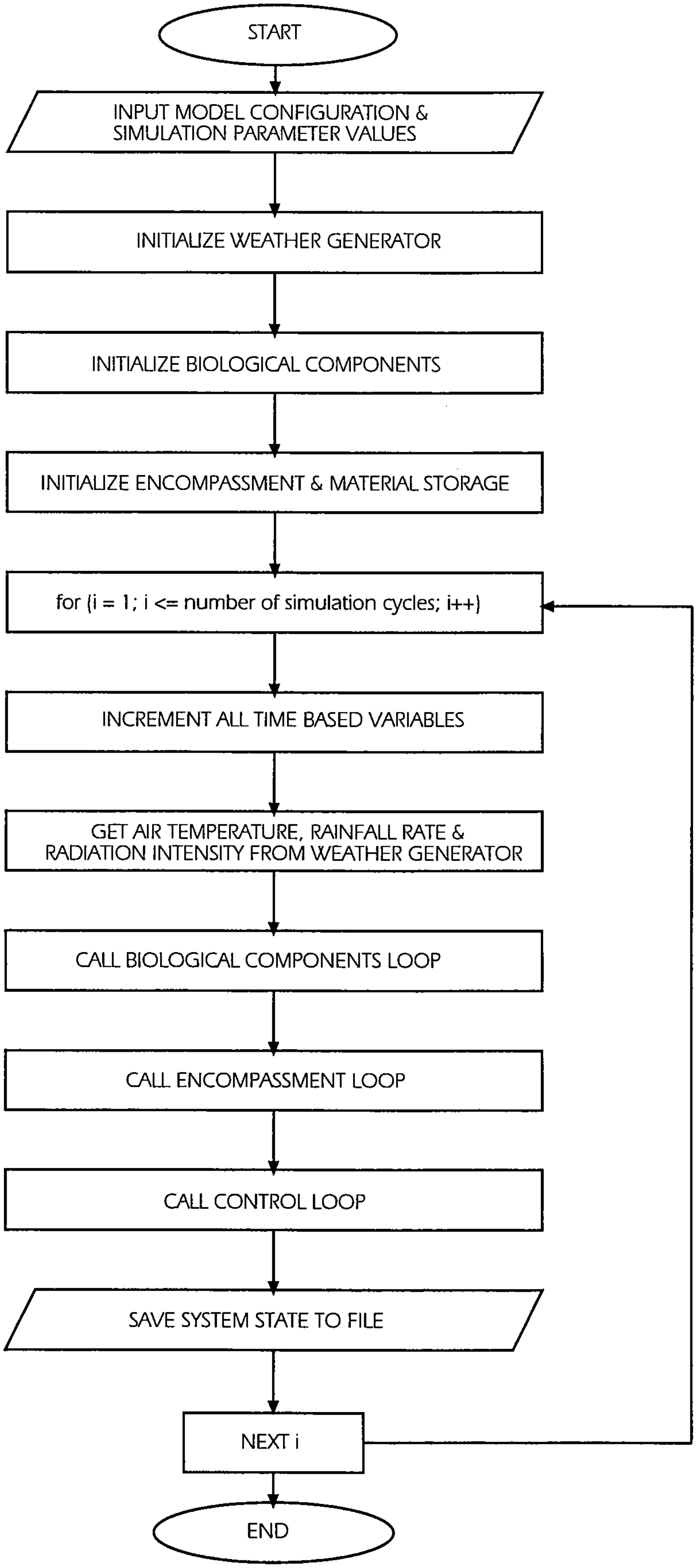


Figure 3.10: Main simulation program loop.

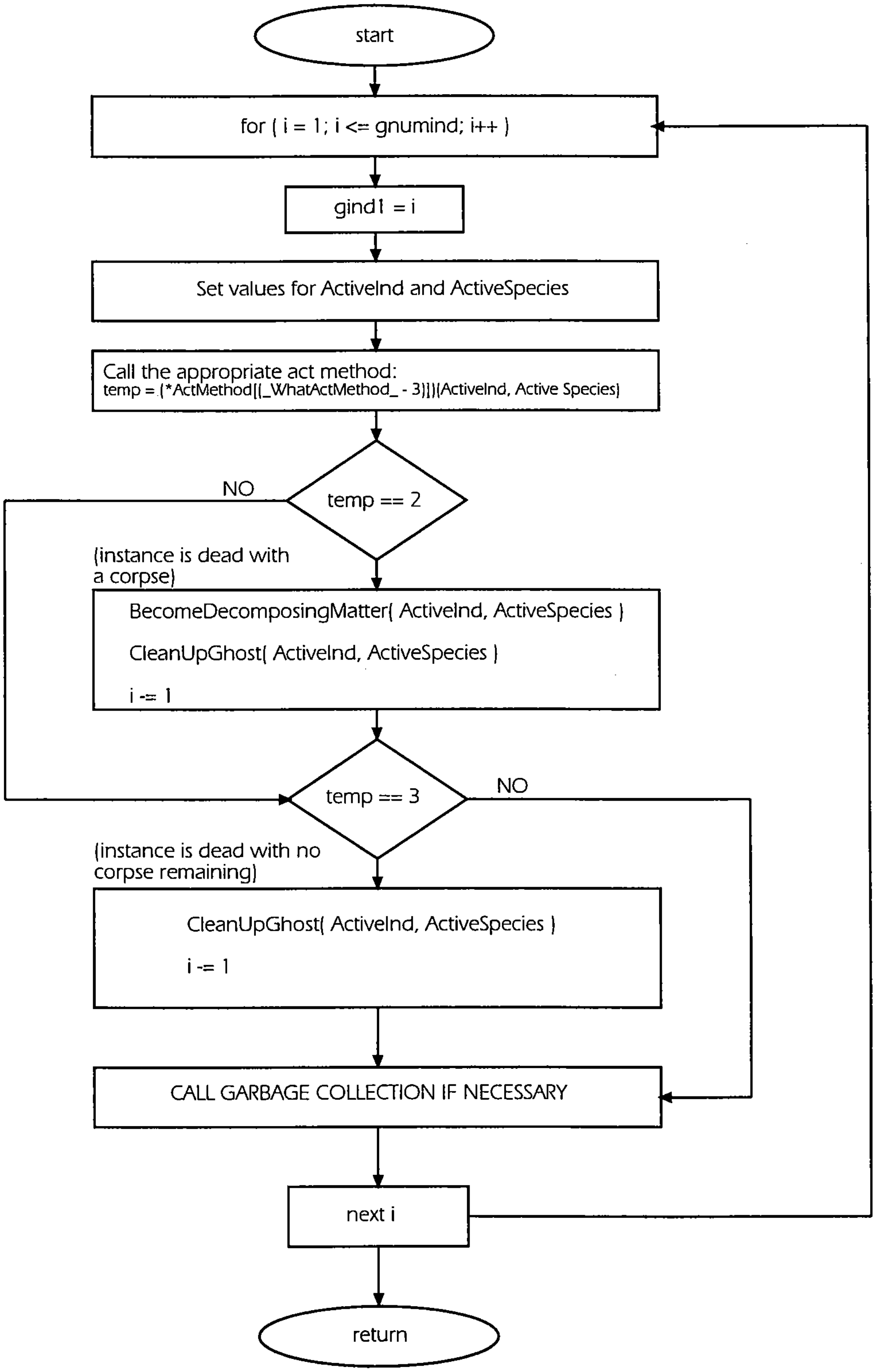


Figure 3.11: Flowchart of the BiologicalComponentLoop routine.

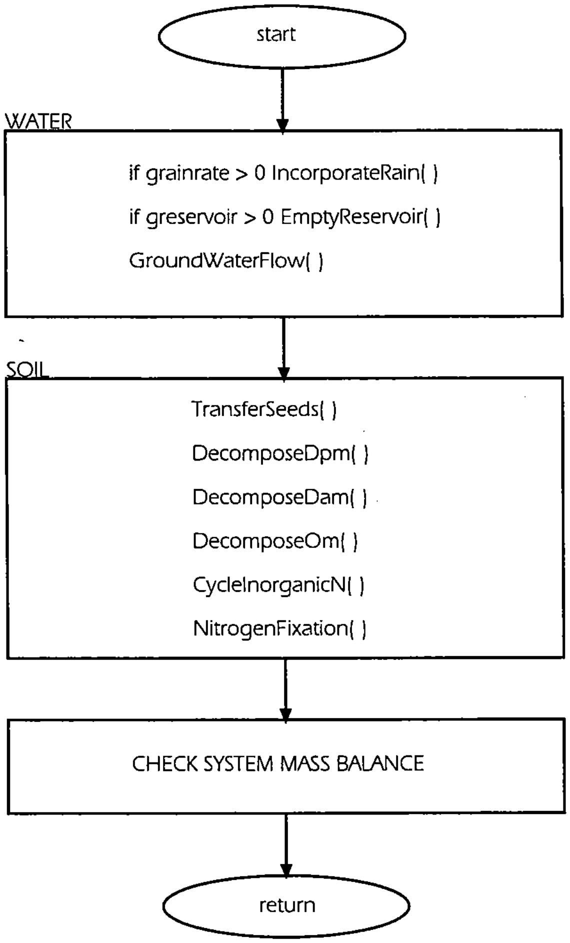


Figure 3.12: Flowchart of the EncompassmentLoop routine.

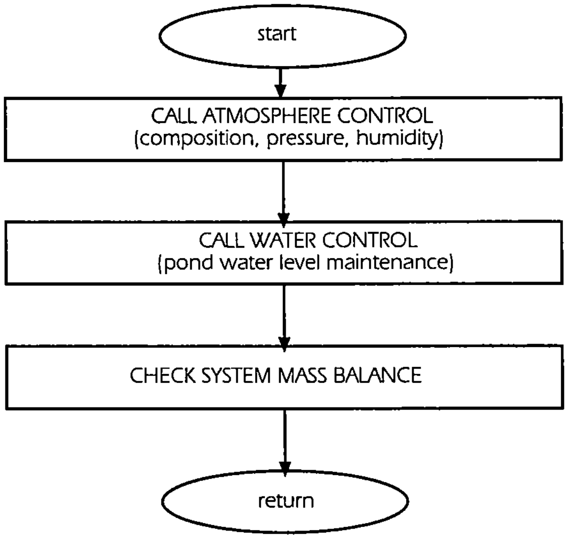


Figure 3.13: Flowchart of the ControlLoop routine.

completed, an integer value describing the status of the instance (alive; dead with no "corpse"; or dead with "corpse" remaining) is returned to the BiologicalComponentLoop routine. If the instance is no longer alive, any of its remaining mass will be added to the soil, and its locations in the various population and index vectors are marked as "garbage" space.

The computational model does not uphold the message sending metaphor of object-oriented programming, in which objects are agents rather than passive data structures. Thus, unlike an object-oriented model, in which objects actively wait for messages to be sent by other objects, and then respond accordingly, in this model, a method is called to perform a task or procedure on a particular instance. Also, polymorphism (in which objects respond differently to the same message by using methods specific to their class) is not supported; in the approach described here, different methods must have distinct names. A kind of polymorphism was implemented for the Act and Grow methods, however. There may be approximately 50 slightly different versions of these. Each instance has two attribute values, corresponding to the "identification numbers" of the Act and Grow method versions that apply to instances of its species. The calling of the appropriate Act or Grow method for an instance is done by retrieving a function pointer from the ActMethods or GrowMethods vector, where the value of the instance's WhatActMethod or WhatGrowMethod attribute determines which element of that vector is accessed (see code in Figure 3.11). All other methods (e.g., Respire, MoveTowards, GiveBirth, etc.) are called directly from within the Act and Grow methods, and are not differentiated on the basis of species class (i.e., the same method version is used for every instance, regardless of species).

As previously mentioned, the retrieval of attribute values for an instance being treated by a method is achieved by computing their locations in the `gpopfloat`, `gpopint` and `gpopchar` vectors using the `gndx001` and `gndx009` indexes. This approach, however, gives rise to fairly lengthy and difficult to decipher source code. The methods were, therefore, written in a pseudo-C code in which there are only symbolic, rather than direct, references to attribute values. For example, a line of pseudo-code might read:

```
_digestedfood_ = _massfoodinstomach_ * _digestrate_
```

where the three symbolic names represent attribute values of the active instance being considered. Before compilation of the simulation program, the psuedo-code is translated into C with the use of a search/replace lookup table. Thus, in C source code, the above code fragment becomes:

```
(gpopfloat[gndx001[ActiveInd][3] + gndx009[ActiveSpecies][32]]) =
        (gpopfloat[gndx001[ActiveInd][3] + gndx009[ActiveSpecies][62]]) *
        (gpopfloat[gndx001[ActiveInd][3] + gndx009[ActiveSpecies][52]])
```

Where "ActiveInd" can be gind1 or gind2 and ActiveSpecies is the species number (value ranging from 1 to gs) of the active instance. (The values of both ActiveInd and ActiveSpecies are sent as arguments to every method.) Although all the attributes in the above example are of the floating point data type, character and integer type attributes are accessed in a similar manner. For instance, (gpopint[gndx001[ActiveInd][2] + gndx009[ActiveSpecies][10]] refers to attribute #10 which is of integer type.

The symbolic referencing of attribute values in this way not only makes the original programming code much easier to read and more accessible, but also avoids the "hard coding" of attribute definitions in the methods, so that a specific ID number is not necessarily permanently bound to a particular attribute. For example, a method that makes use of the leaf area index will always be written in terms of the symbolic name _leafareaindex_. In this way, it becomes possible to make this name correspond to attribute #214 (or any other) rather than #60; which attribute number it actually corresponds to is not important until just prior to compilation. Thus, this allows changes to be made to the attribute and biological class definitions without necessitating alterations to the method code.

After execution of the BiologicalComponentLoop routine, the

EncompassmentLoop routine (Figure 3.12) is called. First, any rainfall that has fallen during the time increment is added to the water content of each grid cell. Next, the water reservoir, which contains all surface runoff due to rain, is emptied into the pond. Lastly, the GroundWaterFlow routine is called to move subsurface water through the soil along the hydraulic gradient. After all the water functions have been dealt with, a number of soil routines are called. The first of these are various decomposition routines that 1) change a fraction of non-viable seeds into decomposing plant matter; 2) decompose fractions of plant, animal and organic matter; 3) transfer some of the inorganic nitrogen from the soil to the atmosphere. Each of these routines is executed for each grid cell individually. Lastly, a nitrogen fixation routine is called, and nitrogen from the atmosphere is fixed and incorporated into the soil. Descriptions of all of the encompassment routines are given in further detail in Parrott and Kok (2000b).

The ControlLoop (Figure 3.13) is the final routine called. It is used to maintain the values of a few critical system variables such as atmospheric pressure and humidity, as well as the level of water in the pond (the pond is rain fed and has no "natural" outlet). The control routines operate by moving materials between the encompassment and material storage realms as required.

## 3.8. Conclusions

The model that has been developed as part of the EcoCyborg Project is novel in that it consists of very detailed, object-based representations of components, and yet encompasses a broad scope, allowing for the representation of all of the major biotic and abiotic parts of an ecosystem. In this sense, it is a unique contribution to both the fields of ecological modelling and ecosystem engineering. The flexibility provided by the configuration programs allows for the possibility of tuning the model to depict many different types of ecosystems, facilitating experimentation and exploration. In addition, the modularity of the code makes it easy to add components and more sophisticated routines. Thus, not only can the model be continually improved upon, it can also be modified to represent, for example, marine or desert environments. The combination of procedural and object-oriented programming techniques used in writing the simulation program proved to be effective, and allowed for the creation of highly portable code. Overall, the simulation program has proven to be sturdy and operative, and will be used with a wide variety of model configurations to study the engineering of ecosystems.

## 3.9. References

Clark, O. G. and R. Kok. 1999. Characterizing biosystems as autopoietic entities. Submitted for publication to *Oikos*.

Clark, O.G., R. Kok, and P. Champigny. 1997. Generation of a virtual terrain. *Environmental Modelling and Software* 12(2/3): 143-149.

Friend, A.D, A.K. Stevens, R.G. Knox and M.G.R. Cannell. 1997. A process-based, terrestrial biosphere model of ecosystem dynamics (Hybrid v3.0). *Ecological Modelling* 95: 249-287.

Fitz, H.C, E.B. DeBellevue, R. Costanza, et al. 1996. Development of a general ecosystem model for a range of scales and ecosystems. *Ecological Modelling* 88: 263-295.

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.

Hogeweg, P. and B. Hesper. 1990. Individual-oriented modelling in ecology. *Mathematical and Computational Modelling* 13(6): 83-90.

Judson, O. 1994. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution* 9(1): 9-14.

Kawata, M. and Y. Toquenaga. 1994. From artificial individuals to global patterns. *Trends in Ecology and Evolution* 9(11): 417-421.

Maturana, H.R. and F. Varela. 1980. *Autopoiesis: The organization of the living.* Dordrecht, Holland: D. Reidel Publishing Co.

McCauley, Edward, William G. Wilson and Andre M. de Roos. 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.

Parrott, L. 1995. The EcoCyborg Project: A model of an artificial ecosystem. M.Sc. thesis, Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal..

Parrott, L., O.G. Clark and R. Kok. 1999. The EcoCyborg Project. Appendix C of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000a. Incorporating complexity in ecosystem modelling. Chapter 2 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000b. Baseline performance of an ecosystem model and simulation for the EcoCyborg Project. Chapter 4 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

## 3.10. Appendix: Simulation program global variables

See next page.

## 3.10 Appendix: Simulation program global variables

| NAME | TYPE | UNITS | DIMENSIONS | DESCRIPTION |
|---|---|---|---|---|
| gyeartime | double | s | 1 | time elapsed in current year of simulation |
| gdaylength | int | n/a | 1 | number of hours of light in present day |
| gelements | double ** | kg | 6x5 | elemental mass balance for the entire system |
| gcompounds | double ** | kg | 10x5 | compound mass balance for the entire system |
| gtotalstartmass | double | kg | 1 | total mass in system at initialisation |
| gmasserror | double | kg | 1 | amount by which the total mass is allowed to deviate |
| BIOLOGICAL COMPONENT VARIABLES AND PARAMETERS | | | | |
| gnumindmax | int | n/a | 1 | maximum number of individuals |
| gnumind | int | n/a | 1 | current total number of individuals |
| gs | int | n/a | 1 | total number of defined species |
| ga | int | n/a | 1 | total number of defined attributes |
| gm | int | n/a | 1 | total number of defined methods |
| gss | int | n/a | 1 | number of selected species |
| gpopint | int * | various | 1xgicountmax | vector of integer attribute values for all instances |
| gpopchar | unsigned char * | various | 1xgccountmax | vector of character attribute values for all instances |
| gpopfloat | double * | various | 1xgfcountmax | vector of double attribute values for all instances |
| gicount | int | n/a | 1 | end of used space in gpopint |
| gccount | int | n/a | 1 | end of used space in gpopchar |
| gfcount | int | n/a | 1 | end of used space in gpopfloat |
| gicountmax | int | n/a | 1 | maximum length of gpopint |
| gfcountmax | int | n/a | 1 | maximum length of gpopfloat |
| gccountmax | int | n/a | 1 | maximum length of gpopchar |
| gattrmat | unsigned char ** | boolean | gsxga | matrix of attribute applicablility by species type |
| gmethmat | unsigned char ** | boolean | gsxgm | matrix of method applicability by species type |
| gattrtype | int ** | n/a | gax3 | table of attribute characteristics |
| gattrvect | int ** | n/a | gsx4 | number of attributes per species (total,char,int,float) |
| gmethvect | int * | n/a | gmx1 | number of methods per species |
| gspeccrea | double * | various | ~1x10^4 | vector of attribute creation values for all species |
| geatprefs | float ** | n/a | gsxgs | eat preferences matrix |

| | | | | |
|---|---|---|---|---|
| gdeadfspace | int | n/a | 1 | amount of dead space in corresponding population vector |
| gdeadispace | int | n/a | 1 | amount of dead space in corresponding population vector |
| gdeadcspace | int | n/a | 1 | amount of dead space in corresponding population vector |
| gdeadspacemax | int | n/a | 1 | max allowable number of dead spaces in population vectors |
| gnumspeccase | int | n/a | 1 | number of “special case” attributes |
| gspeccase | int * | n/a | 1xgnumspeccase | vector of special case attributes |
| gind1 | int | n/a | 1 | position in population index (gndx001) of active individual #1 |
| gind2 | int | n/a | 1 | position in population index (gndx001) of active individual #2 |
| gspecies1 | int | n/a | 1 | species ID of active individual #1 |
| gspecies2 | int | n/a | 1 | species ID of active individual #2 |
| gcanopy | double *** | m^2 | gxcellsxgycellsx3 | total leaf area in each level of the canopy at each grid cell |
| gndx001 | int ** | n/a | gnumindmaxx3 | start positions for each individual in the population vectors |
| gndx002 | | | | |
| gndx003 | | | | |
| gndx004 | int ** | n/a | gdeadspacemaxx2 | index to dead space locations in the 3 population vectors |
| gndx005 | int *** | n/a | gxcellsxgycellsx2 | index to gndx006 |
| gndx006 | int * | n/a | numindmaxx1 | vector of individuals ordered by xy location |
| gndx007 | int ** | n/a | gsx3 | index to gndx008 |
| gndx008 | int * | n/a | gnumindmaxx1 | vector of individuals ordered by species |
| gndx009 | int ** | n/a | gsxga | matrix of offsets for attributes in population vectors |
| gndx010 | int * | n/a | gsx1 | vector of species definition start locations in gspeccrea |
| gndx011 | int ** | n/a | gsx2 | matrix containing the number of instances per species |
| gndx012 | int ** | n/a | gsx1 | matrix of attribute start offsets in the gspeccrea vector |
| gspace6 | int | n/a | 1 | amount of free space to leave at end of each record in gndx006 |
| gspace8 | int | n/a | 1 | amount of free space to leave at end of each record in gndx006 |
| gndx6countmax | int | n/a | 1 | maximum length of gndx006 vector |
| gndx8countmax | int | n/a | 1 | maximum length of gndx008 vector |
| gndx6count | int | n/a | 1 | actual length of gndx006 vector |
| gndx8count | int | n/a | 1 | actual length of gndx008 vector |
| glastid | int | n/a | 1 | value of last assigned individual ID |

## ENCOMPASSMENT VARIABLES & PARAMETERS

| | | | | |
|---|---|---|---|---|
| gxcells | int | n/a | 1 | number of terrain grid cells on the x-axis |
| gycells | int | n/a | 1 | number of terrain grid cells on the y-axis |
| gxcellwidth | double | m | 1 | width of a grid cell on its x-axis |
| gycellwidth | double | m | 1 | width of a grid cell on its y-axis |

| | | | | |
|---|---|---|---|---|
| gstorvol | double | m^3 | 1 | volume of material storage chambers |
| gairvol | double | m^3 | 1 | volume of atmosphere |
| gairpressure | double | Pa | 1 | ambient airpressure |
| grh | double | % | 1 | relative humidity of air |
| gairtemp | double | ˚C | 1 | ambient air temperature |
| grad | double | W·m^2 | 1 | irradiation rate |
| genergy | double | J | gxcellsxgycellsx1 | amount of radiant energy recieved per grid cell |
| gsoilporosity | double | % | 1 | soil porosity |
| gsoilconductivity | double | m·s^-1 | 1 | soil conductivity |
| gsoilN | double | kg | 1 | starting mass of inorganicN in soil |
| gsoilNtrate | double | s^-1 | 1 | rate of transfer of soil inorganicN to atm |
| gNfixrate | double | s^-1 | 1 | rate of nitrogen fixation by soil organisms |
| gomdrate | double | s^-1 | 1 | organic matter decomposition rate |
| gdpmdrate | double | s^-1 | 1 | decomposing plant matter decomposition rate |
| gdamdrate | double | s^-1 | 1 | decomposing animal matter decomposition rate |
| gseedtransrate | double | s^-1 | 1 | rate of seed transfer to decomposing matter pool |
| gelev | double ** | m | gxcellsxgycellsx1 | matrix of terrain elevations |
| gpond | unsigned char ** | boolean | gxcellsxgycellsx1 | matrix of pond/no pond flags |
| gsoil | double *** | kg | gxcellsxgycellsx12 | matrix of masses of each soil and water fraction by grid cell |
| gsoilwater | double *** | kg | gxcellsxgycellsx2 | matrix of max. sat/unsat water mass for each soil cell |
| gdplant | double *** | kg | gxcellsxgycellsx8 | matrix of decomposing plant matter composition |
| gdanimal | double *** | kg | gxcellsxgycellsx8 | matrix of decomposing animal matter composition |
| gom | double *** | kg | gxcellsxgycellsx8 | matrix of organic matter composition |
| gseed1 | double *** | kg | gxcellsxgycellsx8 | matrix of seed composition |
| gseed2 | double *** | kg | gxcellsxgycellsx8 | matrix of seed composition |
| gseed3 | double *** | kg | gxcellsxgycellsx8 | matrix of seed composition |
| gseed4 | double *** | kg | gxcellsxgycellsx8 | matrix of seed composition |
| gseed5 | double *** | kg | gxcellsxgycellsx8 | matrix of seed composition |
| grainnet | double ** | n/a | gxcellsxgycellsx1 | matrix of rain attenuation values |
| grainrate | double | m·s^-1 | 1 | rainfall rate |
| gpondlevel | double | m | 1 | pond water level setpoint value |
| greservoir | double | kg | 1 | mass of water stored temporarily in pond reservoir |
| gpondcells | int ** | n/a | gnumpondx2 | list of pond cell coordinates |
| gnumpond | int | n/a | 1 | number of terrain cells that are pond cells |

### *Connecting text between Chapters 3 and 4*

In Chapter 3, the overall modelling approach used to represent the ecosystem, and the program used to implement it in simulation were described. Whereas, in that chapter, features of the *computational* model were described, the remaining chapters of the thesis have as their focus the *representational* ecosystem model. In Chapter 4, the manner in which the components and processes of the encompassment and material storage realms are represented is described in detail, and the results of three long-term simulations are presented. In the first of these, the model was configured to represent an ecosystem without any biological components. The other two simulations were based on configurations of strictly plant-based ecosystems. The objective in performing these simulations was to assess the operation of major encompassment, material storage, and control routines with and without the presence of biological components, and to observe the system's response to the forcing functions. These simulations, which allow for an evaluation of the overall performance of the model, are referred to as "baseline simulations", and serve as standards by which the results of simulations based on more complex ecosystem constitutions can be compared.

The routines discussed in Chapter 4 are included in the files "encompass.c", "control.c" and "bio.c" and are available for viewing on the enclosed CD-ROM (Appendix B).

Chapter 4 was coauthored by L. Parrott and R. Kok and is currently being prepared to be submitted for publication. An earlier version of the paper was presented as: Parrott, L. and R. Kok. 1999. Baseline performance of an ecosystem model and simulation for the ecocyborg project. Presented July 1999 in Toronto Ont. at the International Meeting of the ASAE, Paper no. 995041. ASAE, 2950 Niles Road, St. Joseph, MI.

# Chapter 4. Baseline Performance of an Ecosystem Model and Simulation for the EcoCyborg Project

## Abstract

The overall objective of the work reported here is to develop and test a computational, object-based model of a materially closed, artificial ecosystem, and to then use the model to study the comportment of different system constitutions with varied initial conditions. In this article, an overview of the model is given, with a detailed description of how different abiotic components and material processes are depicted. Results of three, 70-year simulations with ecosystem configurations that include soil, air and water subsystems, as well as decomposers and primary producers are presented. Experiments based on these simple configurations are being used to establish a record of the baseline performance of the model when implemented in simulation so as to provide a foundation for comparison with later simulations in which ecosystems with much more intricate constitutions will be modelled.

## 4.1. Introduction

The engineering of *ecocyborgs*, ecological systems that are integrated with technological control systems, will have a number of important future applications, especially in areas such as urban design, environmental restoration, and off-Earth exploration and development. Although many types of ecocyborgs, such as automated greenhouses, artificial wetlands and controlled ecological life support units, are currently being engineered, the approach that is followed is often based on mostly experiential knowledge. The long-term objective of our research program is, therefore, to improve ecocyborg engineering to the point that it will be possible to design such systems according to standard principles and methods that are founded on a sound theoretical basis. That is, we aim to develop an appropriate theoretical framework to enable a much more sophisticated approach to the design of ecocyborgs and other similar complex biological networks. Such a framework must provide for both the description and explanation of the relationship between the constitution (composition and structure), initial state, and resultant comportment (dynamics or behaviour) of such systems, when subjected to a given set of forcing functions. For engineering purposes, the inverse of this relationship must also be elucidated so as to make it possible to design (and modify, repair, etc.) these systems to display a particular target behaviour. Ultimately, this will enable a more refined approach to the engineering of all types of ecocyborgs, ranging from marginally controlled, natural habitats to highly autonomous, artificial environments.

These objectives are being pursued as part of the EcoCyborg Project, a research initiative being undertaken at McGill University. In the current phase of the project, a set of software tools with which computational models of ecocyborgs can be formulated and implemented in simulation is being developed. Current modelling efforts are focused on one specific, although hypothetical, setting of an ecocyborg situated in an orbital space station. This setting, although it does not physically exist, has been envisioned such that it is achievable with either present day, or readily-foreseeable technologies, and does not contravene the laws of physics, chemistry, biology, etc., as we understand them. Thus, our approach is to use and configure related models of this hypothetical scenario in order to explore the effects of various forcing functions, starting conditions and constitutions on overall system comportment.

The focus of this article is an ecosystem model, which is an integral part of the software-based ecocyborg engineering "tool-kit". This model has been developed such that it meets the boundary conditions imposed by the space station setting, and can be configured to represent different possible design scenarios. Although it will later be linked to a sophisticated control system (as part of an overall ecocyborg model), its performance in simulation is currently being tested independently (except for some very rudimentary control components that are regarded as being intrinsic to the ecosystem).

While the model does reflect the nature of real ecosystems to some degree, in general, it is not meant to provide an accurate prediction of the behaviour of any particular ecosystem or biogeographic region, but is instead intended to emulate some of the complexity of biological networks, so as to facilitate the study of the relationships between overall system composition, structural characteristics and initial state, and the comportment that emerges. In keeping with this objective, the model consists of a large number of rudimentary components whose individual behaviour or functioning is described in a simple manner. This ensures that each major part of an ecosystem is represented at least to some degree, without exceeding the computational capacity of available hardware during simulation. The model is completely configurable, so that any ecosystem composition (e.g., species types, soil characteristics, topology, climate, etc.), structure (e.g., relationships between components) and initial state can be specified.

The model is presently being tested with limited configurations that include an atmosphere, a terrain, decomposers, and primary producers, but no higher-level consumers. Experiments with these simple configurations are being used to establish a record of the baseline performance of the model when implemented in simulation so as to provide a foundation for comparison with later work in which ecosystems with more intricate constitutions will be modelled. In the sections below, a description is given of the modelled setting and the general approach that has been chosen to represent it. The manner in which the physical environment and the biogeochemical processes are represented is described in detail, followed by the results of some sample "baseline" experiments. These baseline experiments were used to assess the operation of the major abiotic routines in the model, with and without the presence of biological components. The ecosystem initial state specification procedure, and the implementation of the model in simulation are described in Parrott and Kok (2000c). The approach used to model the living components is given in further detail in Parrott and Kok (2000a; 2000b).

### 4.2. Background and literature review

Our intent has been to create a model of an ecosystem that is as comprehensive as possible, including most major living and non-living components, in order to reproduce, to a reasonable degree, the functional complexity of a natural situation. This objective has been addressed by other researchers as well, and there currently exist several fairly comprehensive ecosystem models. Examples are: Hybrid v.3.0 (Friend et al., 1997) which was written to represent the terrestrial biosphere within a larger whole Earth system model, the General Ecosystem Model (GEM) which was developed as a research tool to simulate a variety of ecosystem types (Fitz et al., 1996), and the Terrestrial Ecosystem Model (TEM) which is being developed by a team at the University of New Hampshire as part of NASA's Earth Observing System (on the web at: http://www.eos-ids.sr.unh.edu/). All of these include primary producers (i.e., plants) which are represented in varying degrees of detail, but none of them include higher trophic levels (i.e., animals).

While these models provide interesting descriptive and predictive capabilities with regards to the major terrestrial biogeochemical cycles and climatic influences, they were developed as tools of science, and do not address biosystems engineering questions. In contrast, the EcoCyborg Project ecosystem model is being developed as an engineering research tool, with an emphasis on configurability and the capacity to specify new components. Thus, rather than being descriptive in nature, this model is intended to be prescriptive; allowing for experimentation with different possible configurations or designs, regardless of whether or not these currently exist in physical reality. This degree of configurability has been accomplished, in part, through the use of an "object-based" modelling approach (Parrott and Kok, 2000d) in which every component in the ecosystem is represented as a distinct entity whose functions and interrelationships with other objects are described by simple rules. With this approach, commonly referred to as 'bottom-up' modelling in ecology, components in the ecosystem are represented at a fairly high degree of resolution, and the global system comportment is allowed to emerge as a result of the combined interactions of the many lower-level entities (Grimm 1999; Judson, 1994; Kawata and Toquenaga, 1994). This approach facilitates model configuration, especially the specification of an ecosystem's initial state, since it is easy to add, modify, and remove object types, and the various occurrences of these. In addition, by emphasizing components and their interactions, an object-based model provides for a system representation that more closely matches a general conceptualization of an ecosystem as a complex biological network.

Object-based modelling in ecology has recently increased in popularity, due to the ease with which certain complex features of ecosystems, particularly with regards to population and community-level dynamics, can be synthesized. However, most of the object-based models that have been developed to date are used to study very specific, single population dynamics, such as animal foraging or territorial behaviours (see, for example, the models of Beecham and Farnsworth (1998), Letcher et al. (1998), or Wolff (1994)). Although several somewhat more inclusive models have been written, including the "RAM" model developed by Taylor et al. (1988) and the "Gecko" model (Booth, 1997), the authors are not aware of any large-scale ecosystem modelling projects that have been developed using an object-based approach. The model described herein is a first attempt to broaden the scope of object-based ecosystem modelling.

## 4.3. Setting and model descriptions

### *4.3.1. Overview*

The ecosystem model has been configured to represent a hypothetical scenario of an ecocyborg that is in an isolated, cylindrical space-station which rotates around its central axis (Figure 4.1). This setting was chosen because it is inherently interesting to the authors; it is also, however, a convenient context in which to model an ecosystem, since the location of the system boundary is obvious, and certain conditions concerning the movement of materials across it

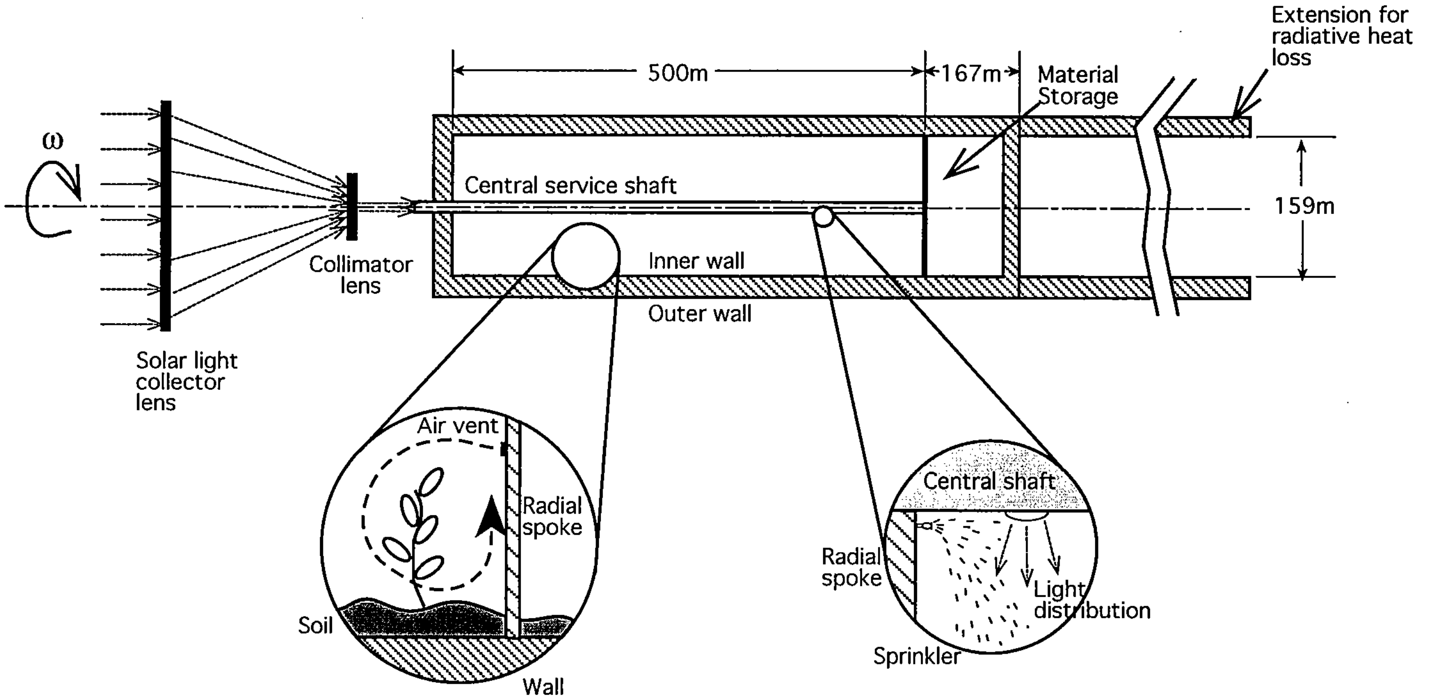


Figure 4.1: The hypothetical ecocyborg (not to scale).

can easily be imposed. Accordingly, the system is assumed to be materially closed, yet open to energy. Radiant energy is presumed to be captured from a nearby star and is used to support primary production in the ecosystem, in addition to meeting any other power requirements for station operation. The ecosystem is situated on the inner walls of the station. It is not production oriented, nor is it specifically intended to support human beings in space[1]. Rather, it is envisioned as being similar in constitution to a natural ecosystem, containing representative species of plants and animals at all trophic levels. At one end of the station, large chambers for the storage of a number of essential compounds are provided. These are intended to compensate for the small size of the ecocyborg by mimicking the "buffering" role that the oceans and atmosphere perform for natural ecosystems on Earth. Certain key variables, such as atmospheric pressure and composition are, therefore, controlled by moving compounds between the ecosystem and the storage chambers. The chambers are regarded as an integral part of the ecosystem, although the movement of material between them and the rest of the ecosystem is regulated.

Although ultimately the intent is to portray and model an ecocyborg in which there are very sophisticated controllers that interact with the ecosystem, in the current implementation, only a few, very rudimentary control components are present. Thus, except for the slight control just mentioned, in this work the ecosystem is presented as operating in "open loop mode", simply reacting to a set of climatic forcing functions (temperature, light intensity and rainfall). In this case, it is assumed that the forcing functions are not influenced by internal ecosystem processes, and are not subject to any manipulation on the part of the control system. This arrangement allows for a realistic depiction of the impact of "unsupervised inputs" (Kok and Lacroix, 1993) although some such inputs to which a natural ecosystem would normally be subjected are not accounted for,

---

[1] Although it is not currently intended to support humans, such a system could be used for this purpose (Dempster, 1993). Since it remains an open question whether or not closed ecosystems of this size can persist for decades or millenia, this model is a step towards learning what components (ecological and/or control-based) and degree of complexity will be required for the construction of human settlements beyond Earth.

e.g., lightning. Although there is no feedback to the forcing functions, for rainfall to occur, water is moved by the control system from storage to the encompassment.

The space station ecocyborg, therefore, serves as a hypothetical setting that is being studied in a virtual environment. It is modelled as being composed of four main parts: an enclosure, an ecosystem, a weather generator and a control system (Figure 4.2). The ecosystem part is represented as being composed of three segments: the biological component, encompassment, and material storage realms. The biological component realm contains all of the living organisms in the ecosystem (plants and animals). The encompassment is the abiotic environment in which the biological components exist, consisting of an atmosphere and a terrain (soil and water). The storage realm serves as a mass reserve of materials and, as described above, can be accessed to control a number of key variables. Each realm is further subdivided into constituent components that may, in turn, be made up of even smaller components, etc. The components at the lowest degree of resolution considered in the model are referred to as "objects".

The overall comportment of the ecosystem is elicited in simulation by computing the activities and functions of all of the objects in each realm. In the biological component realm, each object represents either an individual organism, or a small "lump" of organisms, the choice of resolution depending upon the physical size and fecundity of the species being depicted. In the encompassment realm, the atmosphere is modelled with a single, uniform object, and the landscape is divided up into a spatial grid of terrain objects. The material storage realm is divided into four objects corresponding to mass storage chambers. The manner in which all of these objects are modelled is described in further detail below.

The model has been encoded in computational form, as described in Parrott and Kok (2000c). The computational model is presently implemented in a time-driven, stand-alone simulation program which also includes a weather generation module (Parrott and Kok, 1996) and a few limited control functions.

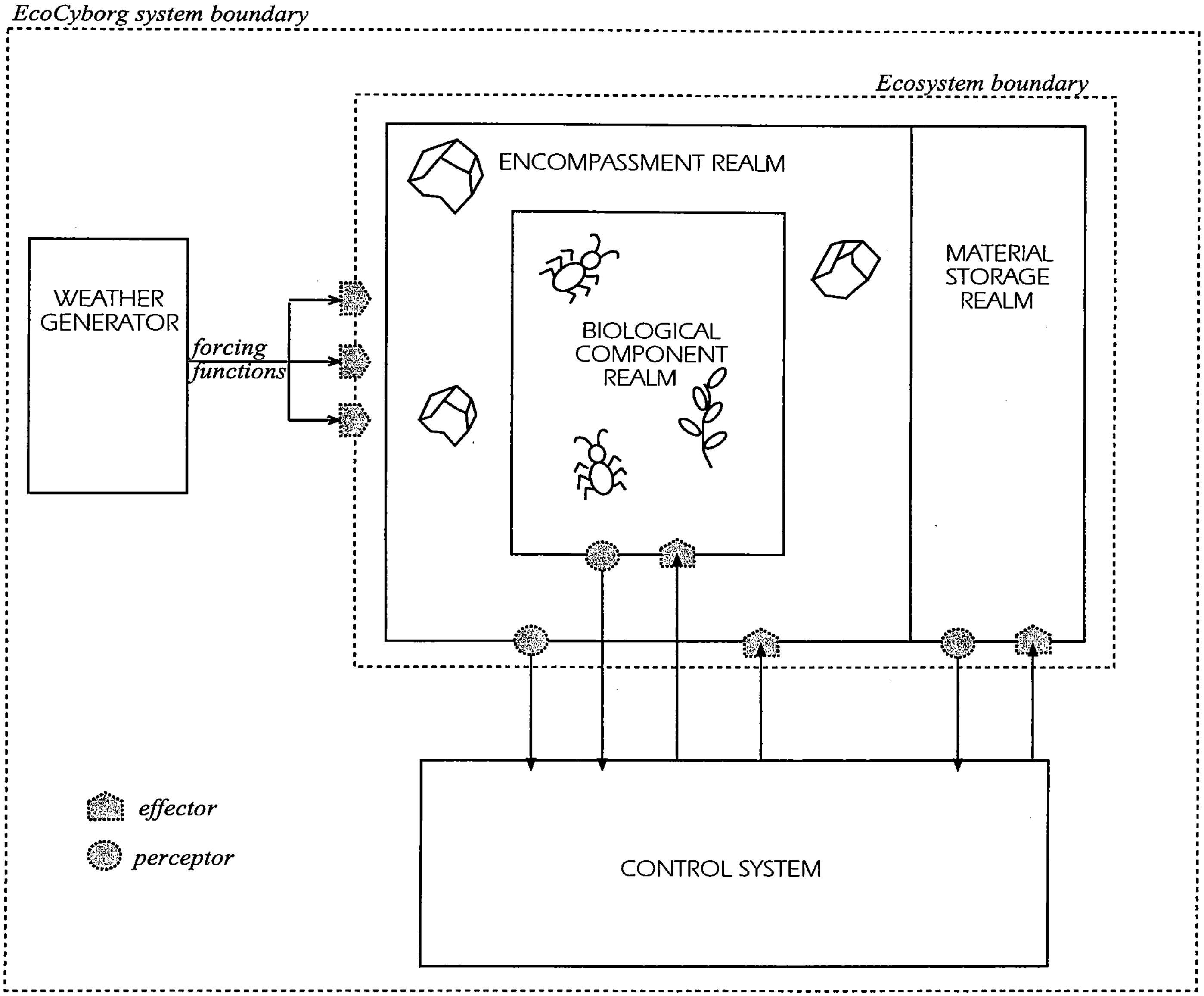


Figure 4.2: Representation of the EcoCyborg.

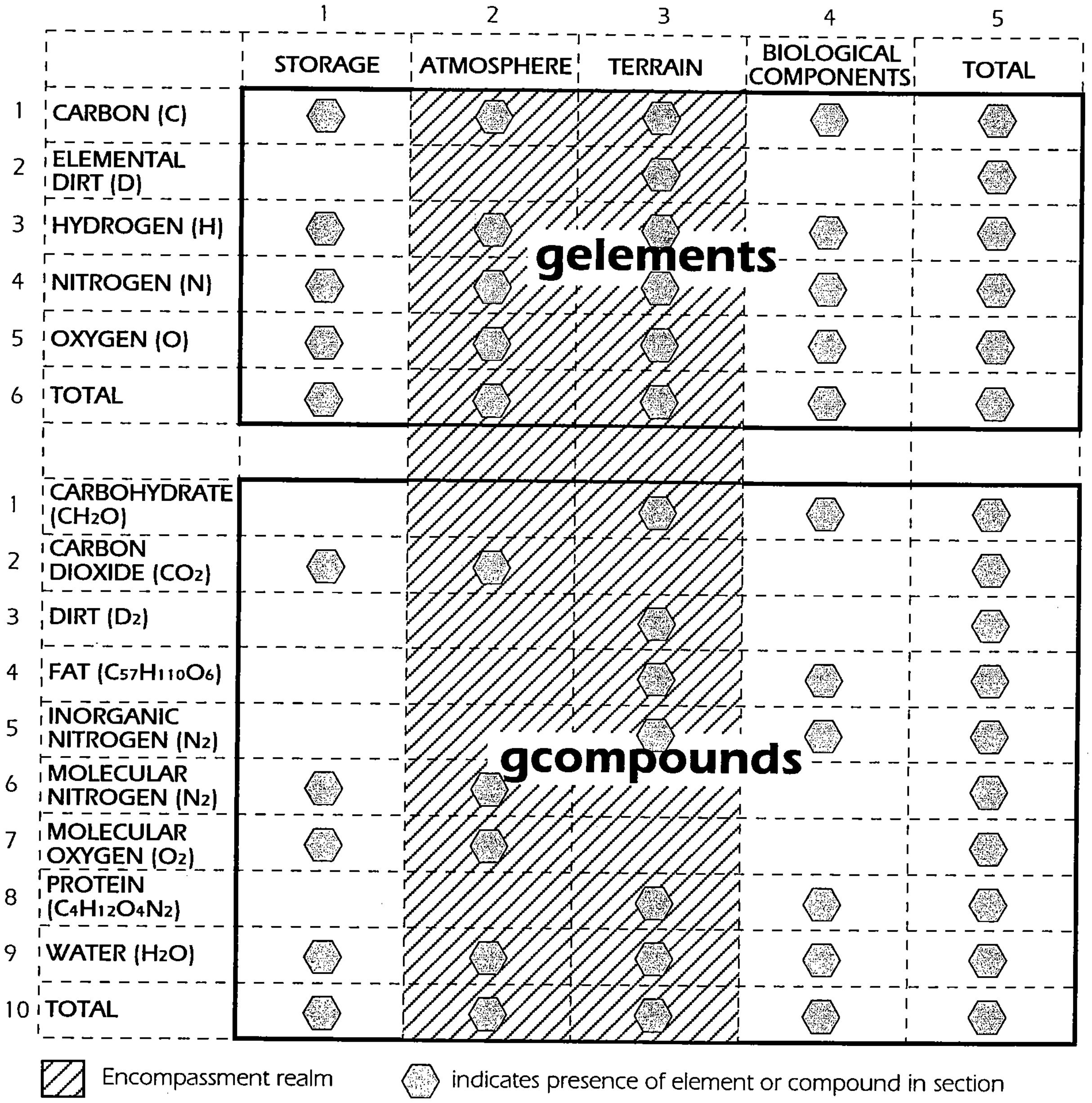
The programming code is written in C and is being compiled and run on desktop computers. Due to the high computational requirements of the model, and the short length of the time step (10 minutes) at which it is iterated, experimental simulations with strictly plant-based ecosystems are currently running at speeds of about 8-12 simulated years per day of machine time (450 MHz Pentium III; 256 Mb of memory; Windows 98), depending upon the number of objects being modelled.

### 4.3.2. *Mass forms and mass accounting*

In the model, all objects have a number of mass and composition attributes or properties, and the interactions between objects largely consist of mass exchanges. Over the course of a simulation, complete mass accounting is done for the ecosystem and every mass exchange, no matter how small, is taken into consideration. As described below, all of the matter in the ecosystem is dealt with and kept track of in parallel in a number of forms. The two forms that are universal throughout the system, and for which detailed system-wide mass accounting is done, are elements and compounds. This allows for a very flexible and fairly realistic modelling approach in which objects of very different types from the three realms can easily interact in a variety of ways, e.g., food relationships as well as respiration and decomposition, etc. Although the metabolism of the primary producers (plants) is entirely driven by radiant energy intensity, and in the consumers (animals) processes such as growth and digestion are partially derived from energy-based considerations, there is no energy accounting done in the model, neither for the whole system, nor for individual objects. Thus, incident radiant energy causes mass conversion and storage in primary producers, and subsequent chemical energy conversion, storage, exchange, etc., is all dealt with in terms of mass conversion, storage and exchange. Thermal energy is assumed to be supplied, absorbed, moved, and dissipated as required.

In order to make the mass accounting approach tenable, and yet to retain a reasonable resemblance to physical reality by allowing differentiation between types of materials, the ecosystem was modelled as consisting of 5 elements, from which 9 compounds may be formed. The elements are: carbon, hydrogen, nitrogen, oxygen, and a fictitious "elemental dirt". Dirt is found only in soil (in its "compound" form, $D_2$) and is an immobile substance that provides structure and volume in the terrain. From the other four elements, the following eight compounds can be formed: carbohydrate, carbon dioxide, fat, inorganic nitrogen, molecular nitrogen, molecular oxygen, protein and water. Although inorganic nitrogen and molecular nitrogen are both considered to have the same composition ($N_2$), they are treated as distinct materials that occur in different parts of the ecosystem. Carbohydrate, fat and protein are each dealt with in terms of a single molecular structure that is assumed to be reasonably representative of the myriad of forms that are found in the physical world. Thus, stearin ($C_{57}H_{110}O_6$) was chosen as a representative fat, and glycine ($C_4H_{12}O_4N_2$) as a representative protein. The distribution of the elements and compounds among the realms etc., is shown in Figure 4.3. During a simulation, mass accounting is done individually for each of the five elements and for the nine compounds, as well as for the totals of all the elements and of all of the compounds. This serves two purposes: First, it allows experiments to be performed with which the effects of material closure on various ecosystem constitutions can be studied and, secondly, it facilitates the verification of the model since any errors are likely to manifest themselves as inaccuracies in the elemental mass balance.

Every object in the ecosystem is made up of some combination of the nine compounds. The atmosphere is composed of molecular oxygen, molecular nitrogen, carbon dioxide and water. Terrain objects contain dirt, carbohydrate, protein, fat, water and inorganic nitrogen (these are distributed among the various types of materials, or mass forms, such as decomposing material and organic matter that make up the terrain). Plants consist of protein, carbohydrate, water, and a small amount of inorganic nitrogen; animals consist of fat, water, and protein. The material storage chambers contain buffers of water, carbon dioxide, molecular nitrogen, and molecular oxygen. These are considered to be



| | | 1<br>STORAGE | 2<br>ATMOSPHERE | 3<br>TERRAIN | 4<br>BIOLOGICAL COMPONENTS | 5<br>TOTAL |
|---|---|---|---|---|---|---|
| 1 | CARBON (C) | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |
| 2 | ELEMENTAL DIRT (D) | | | ⬡ | | ⬡ |
| 3 | HYDROGEN (H) | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |
| 4 | NITROGEN (N) | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |
| 5 | OXYGEN (O) | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |
| 6 | TOTAL | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |

| | | 1<br>STORAGE | 2<br>ATMOSPHERE | 3<br>TERRAIN | 4<br>BIOLOGICAL COMPONENTS | 5<br>TOTAL |
|---|---|---|---|---|---|---|
| 1 | CARBOHYDRATE ($CH_2O$) | | | ⬡ | ⬡ | ⬡ |
| 2 | CARBON DIOXIDE ($CO_2$) | ⬡ | ⬡ | | | ⬡ |
| 3 | DIRT ($D_2$) | | | ⬡ | | ⬡ |
| 4 | FAT ($C_{57}H_{110}O_6$) | | | ⬡ | ⬡ | ⬡ |
| 5 | INORGANIC NITROGEN ($N_2$) | | | ⬡ | ⬡ | ⬡ |
| 6 | MOLECULAR NITROGEN ($N_2$) | ⬡ | ⬡ | | | ⬡ |
| 7 | MOLECULAR OXYGEN ($O_2$) | ⬡ | ⬡ | | | ⬡ |
| 8 | PROTEIN ($C_4H_{12}O_4N_2$) | | | ⬡ | ⬡ | ⬡ |
| 9 | WATER ($H_2O$) | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |
| 10 | TOTAL | ⬡ | ⬡ | ⬡ | ⬡ | ⬡ |

▨ Encompassment realm

⬡ indicates presence of element or compound in section

Figure 4.3: Material composition of the ecosystem as tabulated by the mass accounting matrices, gelements and gcompounds.

stored in the solid state and converted between states as required. Due to the peculiarities of the space station setting, this is a particularly convenient approach; as mentioned, the details of implementation, and the energy requirements needed to accomplish this are not considered here.

Thus, in the model, all the major flows of material that would be expected to occur in the equivalent physical setting are represented and dealt with by means of mass accounting. Ultimately, mass can always be tracked at the elemental level (and must always balance at this level on a system-wide basis). For instance, a carbon atom from the atmosphere might be absorbed and incorporated into plant biomass, next be assimilated into an animal, then become part of the organic material in the soil, and ultimately be returned to the atmosphere via decomposition. All of these transformations would occur as the result of interactions among individual objects residing in the different ecosystem realms. The main avenues of mass flow in the system are illustrated in Figure 4.4.

### 4.3.3. *The encompassment realm*

#### 4.3.3.1. The terrain

The projected area of the ecosystem's ground surface corresponds to the area of the inside wall of the cylindrical space station in which it is enclosed. When "unrolled", therefore, the terrain is rectangular in shape, wrapping in one direction and having impermeable walls on two sides. It is currently modelled as a rectangular array of grid cell objects, each of which has a number of properties such as elevation, water content, quantity of organic matter, etc., see below. These properties are modified during a simulation by the three main terrain processes: decomposition, nitrogen fixation, and subsurface water flow, as well as by the forcing functions (e.g., precipitation) and by the activities of the biological component objects. The terrain dimensions and topology, as well as the spatial resolution with which it is represented are completely configurable, and can, therefore, be specified differently for different simulations. Although

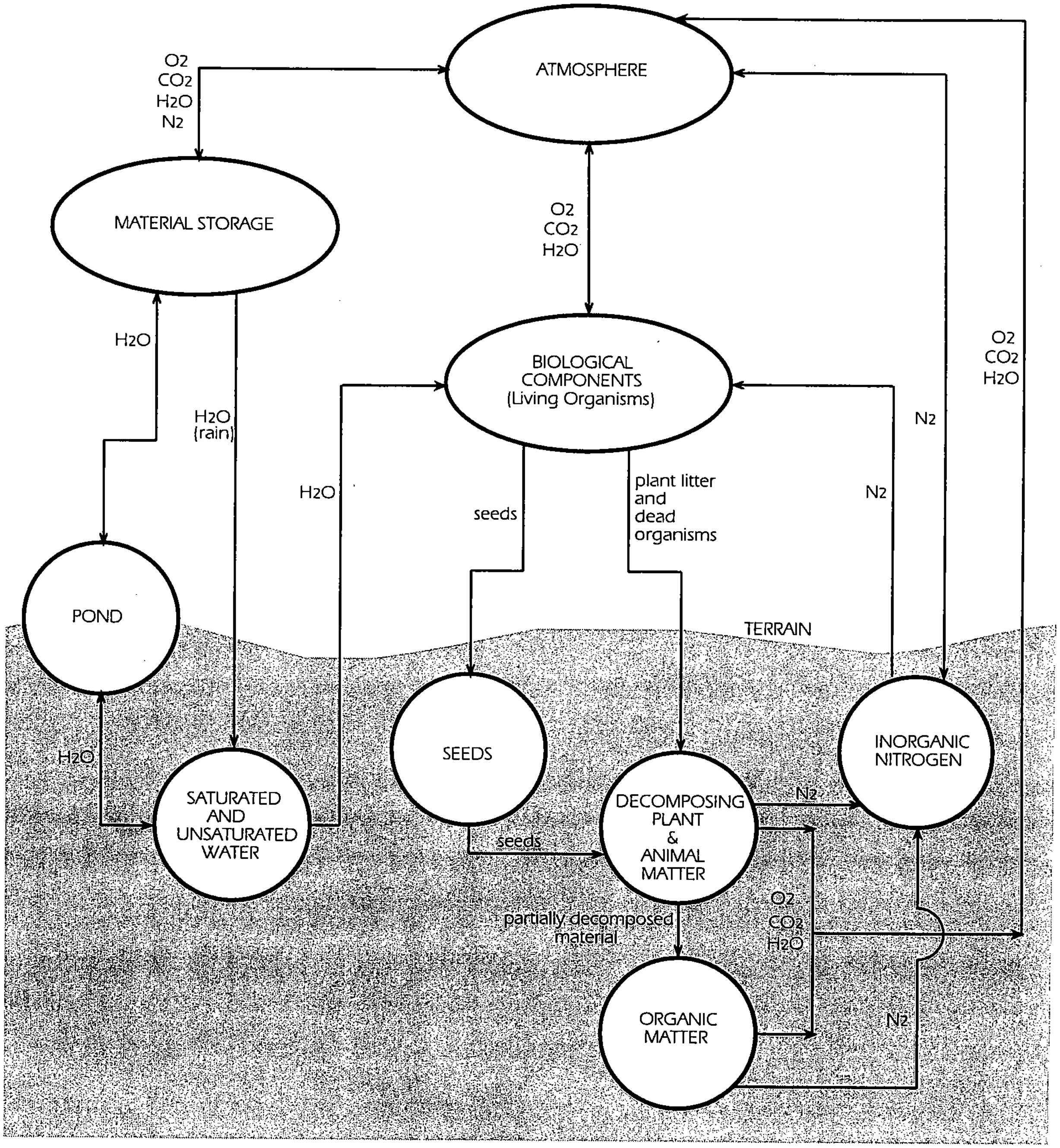


Figure 4.4: Main mass flows in the modelled ecosystem.

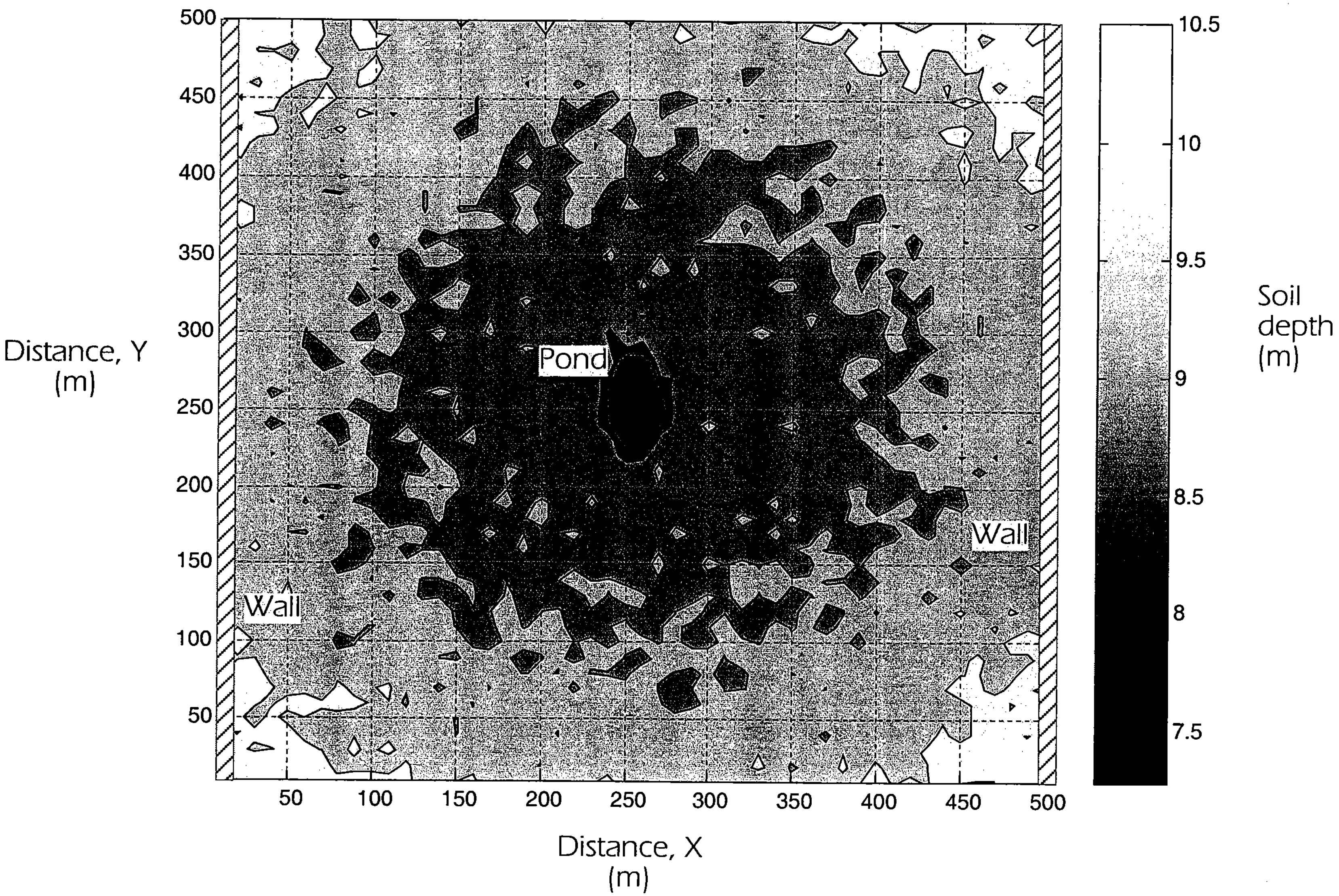


Figure 4.5: Terrain topography currently in use.

the structure of the ecosystem model places no explicit limit on the size of the terrain, in practical terms, when the model is used in simulation, the largest area that can be accommodated at a specific resolution is limited by both the machine memory and the computer's clock speed. For current simulations, a 500m x 500m terrain is modelled, at a resolution of 10m along both axes, resulting in a total of 2500 grid cells. The terrain topography that is currently being used is shown in Figure 4.5; it is a gently sloping landscape with a central pond. It was generated with the program described in Clark et al. (1997).

The distribution of material over the terrain is modelled at the resolution of the grid cells. Each grid cell is treated as a distinct object which is composed of two main substances: soil and water. Water in the terrain is accounted in two parts: the amount of water held in the soil up to field capacity, which is referred to as unsaturated water, and any additional water held in the soil pores or above the soil surface, which is referred to as saturated water. The water table level is calculated from the amount of saturated water. Soil, in turn, is modelled as being composed of the following materials: dirt, inorganic nitrogen, seeds, organic matter, decomposing plant matter and decomposing animal matter. Each of these materials is made up of appropriate combinations of the nine compounds mentioned previously. The masses of all of the soil and water parts, as well as the total mass of each constituent compound and element, are stored as grid cell properties.

Dead plant and animal materials in the soil that are up to 50% decomposed are called decomposing plant matter and decomposing animal matter, respectively. These accumulate in the soil as a result of direct mass transfers from the biological components realm. Decaying material of any origin that is more than 50% decomposed is called organic matter. As discussed below, the decomposition process is one source of the inorganic nitrogen that is present in the soil, nitrogen fixation by soil organisms and by plants being the other. Seeds that are dispersed by plants, but never provided an opportunity to germinate, make up the soil seeds (the viable seeds are modelled as biological components).

These are divided into five different classes, as distinguished by size (Table 4.1), and are available for consumption by gramnivorous species. The five types are treated as independent properties of each grid cell object, and are dealt with separately in terms of decomposition, consumption, etc.

| Seed Size Class | Diameter (mm) |
|---|---|
| 1 | <1 |
| 2 | 1-5 |
| 3 | 5-20 |
| 4 | 20-50 |
| 5 | >50 |

Table 4.1: Categories of seeds

Although the elevation of the terrain could be a function of the quantities of the various materials present, in the present case, since the soil is mostly dirt, its thickness is calculated based only on the volume of dirt present. Thus, the soil depth is always kept equal to the elevation, as determined by the topography, which is assumed to be constant. Soil porosity is also assumed to remain constant, and to be the same throughout the terrain. Various types of decomposers, including micro-organisms, are included implicitly in the soil, their activity being modelled with the overall decomposition process. Nitrogen fixing organisms are also implicitly included.

The decomposition process, in which decaying plant and animal matter is converted to atmospheric compounds and inorganic nitrogen, is depicted in Figure 4.6. The process is divided into a number of stages which are carried out in turn for each terrain object in sequence during every simulation time cycle. The various factors and rate constants for the sub-processes in the different stages are specified prior to the simulation, using a set of configuration programs (as described in Parrott and Kok, 2000c). All the sub-processes are assumed to

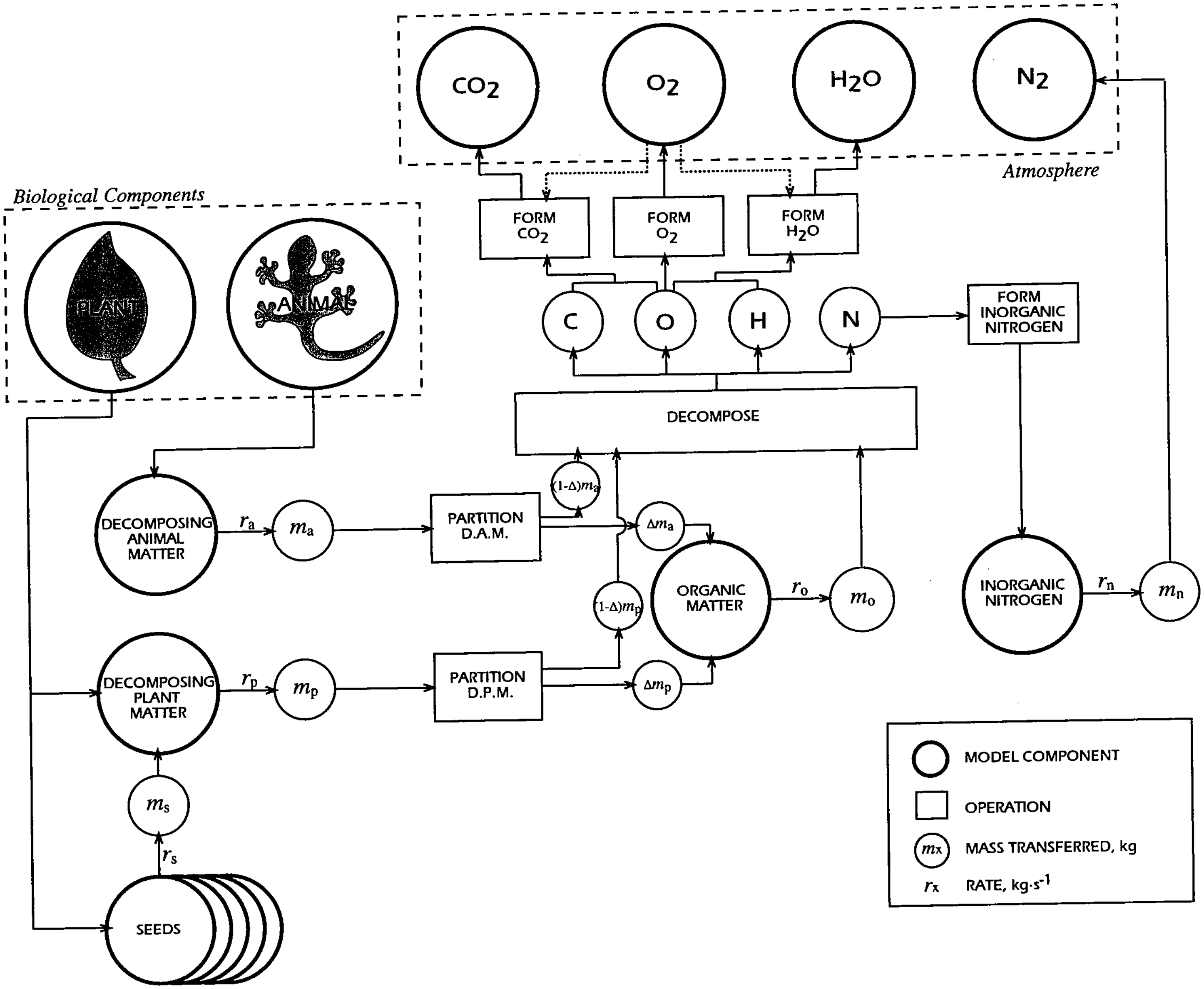


Figure 4.6: The decomposition process.

proceed according to first-order kinetics. In the first stage, fractions of the five seed masses are transferred to the decomposing plant matter pool (five values of $m_s$ calculated by multiplying the seed masses with the value of $r_s$ and the time increment). Next, fractions of decomposing plant and animal matter are removed ($m_a$ and $m_p$) and a portion of each of these is then transferred to the organic matter pool, the remainder being decomposed into atmospheric $CO_2$, $O_2$ and $H_2O$, and into soil inorganic nitrogen, $N_2$. A fraction of the organic matter ($m_o$) is then removed and decomposed in the same manner. Lastly, some of the inorganic nitrogen is returned to atmospheric $N_2$. All of the transfers of mass in the various forms are implemented as adjustments to the properties of the grid cell objects. Although the processes and rates are identical for each grid cell, the cells very quickly become differentiated as the result of unequal inputs of decomposing matter. At this time, the decomposition process is assumed to be independent of temperature or other soil conditions such as moisture content. As previously mentioned, the presence of decomposer organisms that facilitate this process is implicit and no aspect of their life cycles or population sizes is explicitly modelled.

In addition to being created as a result of the decomposition of decaying plant and animal material, inorganic nitrogen is replenished via two avenues of nitrogen fixation. The first is fixation by micro-organisms in the soil. The presence of these organisms is modelled implicitly, and the process by which nitrogen is made available from this source is assumed to follow zero-order kinetics with a rate constant that is a function of temperature (the soil is assumed to be at air temperature); the rate of nitrogen fixation is then calculated based on a reference rate specified for 20°C[1]. The process is assumed to cease at, and below, temperatures of 0°C. As was done for decomposition, this process is also applied during every simulation cycle to each grid cell object in sequence. The resultant amount of fixed nitrogen is removed from the atmosphere and added to the soil's inorganic nitrogen pool. The second source of inorganic nitrogen is fixation by selected plant species in combination with symbiotic soil micro-

---

[1] See article appendix.

organisms. These plants transfer atmospheric nitrogen to the soil during the growth season at a rate that is a function of the plant root mass (for details see Parrott and Kok, 2000a). Thus, the presence or absence of these species in a certain location may lead to differentiation between grid cells with regards to nitrogen availability. Inorganic nitrogen in the soil is reduced by all species of plants because they use it in protein synthesis for growth.

Water can accumulate in a grid cell as the result of surface infiltration and subsurface flow. Water can also leave a cell via subsurface flow. (It may also be removed due to absorption and transpiration by plants.) Since the soil surface was expected to be permanently covered by a dense canopy, direct evaporation from the soil surface is not taken into account. As the quantity of water in a grid cell increases, the unsaturated volume is filled before the saturated volume. When the unsaturated volume is only partly filled, the water table level is zero; when the saturated volume is partly filled, the water table level is between zero and the surface elevation of the cell; when there is more water present than these two capacities combined, the water table level has a value greater than the elevation of the surface, i.e., the cell is flooded. When water is removed from a cell, the excess surface water (if present) is removed first, and then the saturated volume is emptied completely before the unsaturated volume is touched. It is assumed that all precipitation reaches the soil surface (i.e., interception by plants is ignored) and that there is 100% infiltration until the saturated volume is filled. If the soil in a grid cell becomes completely saturated, any excess water is assumed to flow directly into the pond (there is no surface flow model per se; surface water simply travels instantaneously from any location on the terrain to the pond). Of course, this has the effect that, after a heavy rainfall, the level of the pond rises so that the area around the pond becomes flooded. The pond level is, however, controlled, so that when flooding occurs, water is transferred from the pond to the appropriate material storage chamber. Conversely, if the pond water level falls below a certain desired level, the control system adds water from material storage.

In the model, a simple, linear, non-iterative scheme is employed to predict the flow of saturated water through the soil. In this scheme, the amount of water transferred between neighbouring cells during a time interval is directly related to the difference in water table levels at the start of the simulation cycle (as well as the hydraulic conductivity, the size of the area between adjoining cells, etc.)[1]. The amount of saturated water at the end of the interval is calculated for each cell during every simulation cycle based on the flows predicted by differences in initial water table levels. The direction of water flow on the grid is illustrated in Figure 4.7. Due to the presence of the impermeable walls on two sides of the terrain, there are minor edge effects. Overall, however, the water flow pattern produced by this model gives a reasonable representation of groundwater movement, provided that the simulation time interval is sufficiently small. The 600 second interval used currently seems to be adequate in this respect.

#### 4.3.3.2. The atmosphere

The total atmospheric volume is constant and is equal to the volume of the cylindrical space station that is allocated to the ecosystem (i.e., not including the central manifold), minus the volumes of the terrain and of the material storage chambers. The space filled by the other ecosystem components is considered to be negligible relative to the total volume and is ignored in the calculations. The atmosphere is composed of nitrogen, oxygen, carbon dioxide and water. Nitrogen, oxygen and carbon dioxide are maintained in the same proportions as in terrestrial air (respectively 79.0226, 20.946, and 0.0314 mole % of bone-dry air) and the relative humidity is kept at 40%. The total pressure is kept at 101.3 kPa. All the atmospheric control is achieved via transfers of the various compounds between the atmosphere and the material storage chambers. Temperature fluctuations (as imposed by the Weather Generator which produces a different sequence of temperatures for each subsequent year) are a major cause of disturbance, tending to result in pressure fluctuations and relative humidity shifts. Other disturbances to the atmosphere are caused, for example, by various

---

[1] See article appendix.

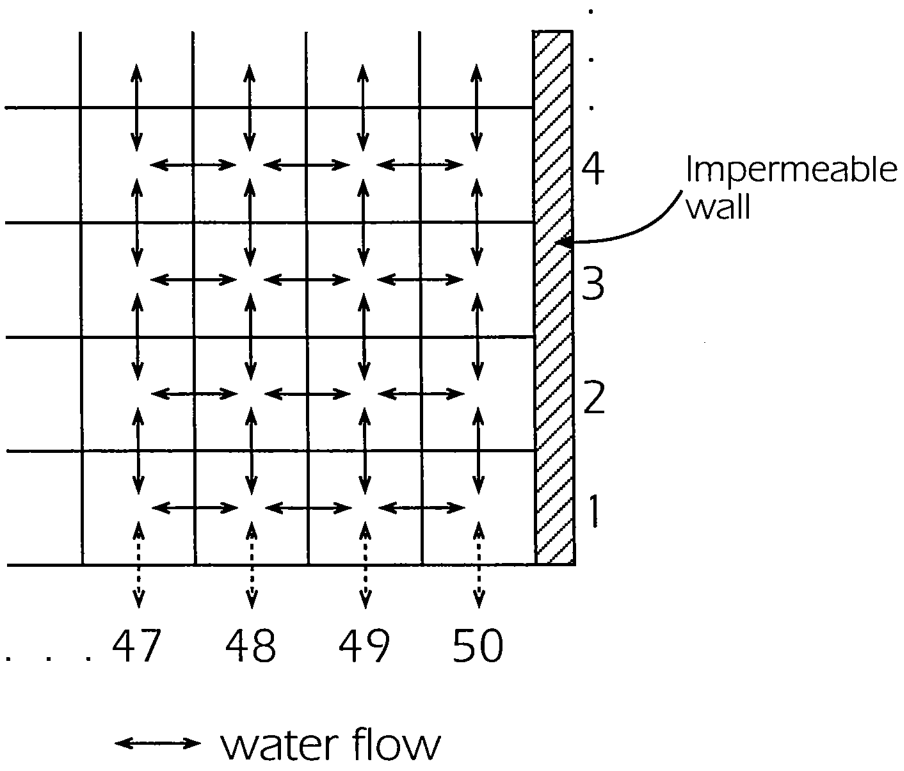


Figure 4.7: Direction of subsurface water flow between terrain cells.

ecosystem processes such as plant photosynthesis, decomposition, nitrogen fixation, etc. The model is based on the assumption that the atmosphere is perfectly mixed, even in the plant canopy layers. It is therefore represented with a single object whose attributes reflect the atmosphere's composition.

### 4.3.4. *The biological component realm*

As previously mentioned, each object in the biological component realm represents either a single individual or a small "lump" of individuals. In order to limit the amount of machine memory resources required for a simulation, lumped objects are limited to one instance per grid cell, covering a maximum area equal to the area of the cell. The model accommodates objects representing a wide variety of different plant and animal species, allowing for the configuration of ecosystems with fairly intricate, multi-trophic level food webs.

Large plants such as trees and bushes are modelled on an individual basis; herbs and grasses are modelled on a lumped basis. Each plant is assumed to be composed of several parts: leaf, stem, root, seed, organic and inorganic storage. A plant grows, becomes senescent, and reproduces according to seasonal cycles. These processes are modelled as being functions of radiation intensity, temperature, and available material resources. Plants compete with one another for light within a canopy as well as for resources such as water and nutrients. The plant objects drive many of the material flows within the ecosystem through the exchange of compounds with the atmosphere due to photosynthesis and respiration, the absorption of water and inorganic nitrogen from the terrain, and the return of organic material to the soil as a result of senescence or death. Animals are also modelled with both an individual and lumped approach. Animal objects, unlike plants, are mobile, and rely on other organisms for food. Each animal grows, metabolizes, ingests and digests food, reproduces, and interacts with other animals, according to a rule-based script. Animals also exchange matter with other components in the ecosystem, for example, gas is exchanged with the atmosphere as a result of respiration, other biological components are consumed due to predation, and material is returned to the soil as a result of excretion or death. In the current version of the model, neither plants nor animals have the capacity to adapt or evolve. Further details regarding how the biological components realm is modelled are given elsewhere (Parrott and Kok, 2000a; 2000b).

### *4.3.5. The material storage realm*

As previously mentioned, the material storage realm is modelled as being composed of four chambers, each of which has one attribute, namely the mass of material it contains. The four chambers contain water, nitrogen, carbon dioxide, and oxygen in solid form. All four of these materials are transferred by the control system to and from the atmosphere as required to maintain pressure or humidity levels, and terrain water is added to and removed from material storage to compensate for rainfall or flooding. The contents of the material storage chambers are not available to biological components, and are only accessed for the few purposes just stated.

## **4.4. Configuration of the model for the baseline simulations**

To obtain an initial assessment of the model's performance and an idea of how the various encompassment processes are affected by the presence of biological components, a number of "baseline" simulations were completed. Three of these are presented here. In this section, the constitutions and initial conditions that were specified for the three cases are described; the results of simulations based on the three ecosystem configurations are presented in the following section.

Parameters for the Weather Generator were set to create mild weather with daily and annual cycles that mimic those of a terrestrial climate in the northern hemisphere. Every year of weather data was slightly different; a sample year is plotted in Figure 4.8. The temperature was set to vary between 0 and 38°C in a normal year, and the total annual rainfall was between 0.8 and 0.9 $m \cdot y^{-1}$. All three simulations were run using the same series of forcing functions (i.e., the

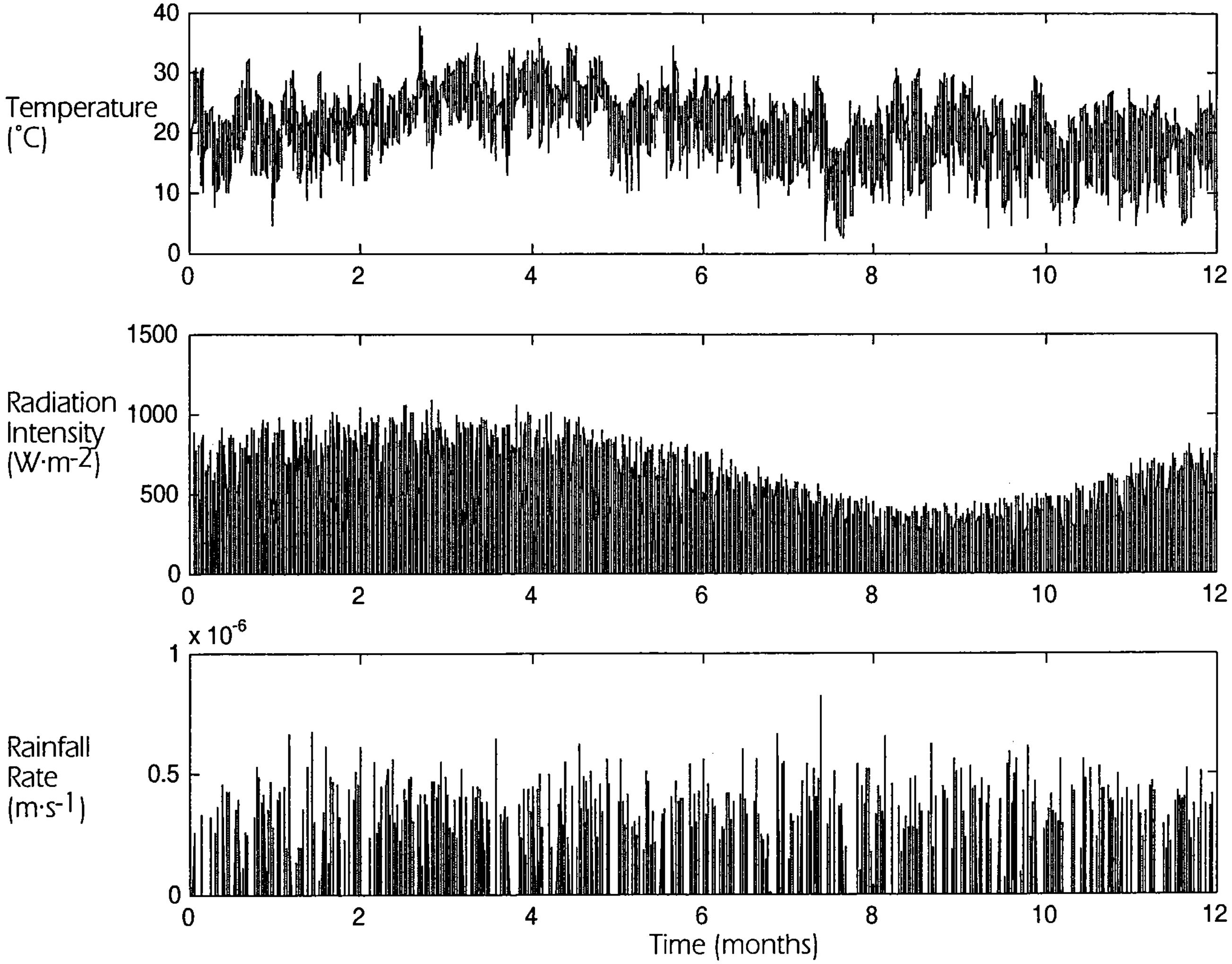


Figure 4.8: Sample weather data for one year as produced by the Weather Generator.

same parameter values, and the same random number generator seed, were supplied to the Weather Generator for all three simulations).

Values of various simulation and model parameters used for the three baseline experiments are given in Table 4.2. Each simulation was run for approximately 75 simulated years[1]. The simulations were started on day 92 of the Julian year, which corresponds to the beginning of the growth season in the ecosystem (t=0 is equivalent to 00:00 on January 1). The three ecosystems were configured and initialised with identical encompassment and material storage realms, but different biological component realms. Initial values of selected encompassment and material storage variables are presented in Table 4.3. The atmosphere was configured to have a standard terrestrial make-up and pressure. The soil was initialized with decomposing matter, organic matter, seeds and inorganic nitrogen distributed uniformly across the terrain. The initial water table level was also uniform. The first simulation (Baseline1) was configured not to have any biological components at all. This was intended to provide for a "control" situation in which the performance of the various encompassment and material storage functions could be assessed in isolation, and with which the performance of other simulations could be compared. For the second simulation (Baseline2), the biological component realm was configured to represent a grassland-type ecosystem composed of six herbaceous species. For the third (Baseline3), the same six herbaceous species were included, with the addition of two tree species. The initial population sizes of each species for the two simulations are given in Table 4.4. For each simulation, the plant instances were initialized with different attribute values selected according to predefined distribution functions (see Parrott and Kok, 2000d).

---

[1] **Baseline1** - Hardware: PowerPC 601 processor, 90Mhz, 84Mb RAM; Operating system: Macintosh OS v8.5; Average execution time: approx. 10 simulated years per day of real time.

**Baseline2 & Baseline3** - Hardware: Pentium III processor, 450Mhz, 256Mb RAM; Operating system: Windows98; Average execution time: approx. 11 simulated years per day of real time.

| Parameter Name | Value |
|---|---|
| SIMULATION CONSTANTS | |
| $\Delta t$ | 600 s |
| Start time | 7862400 s |
| ATMOSPHERIC CONSTANTS | |
| Atmospheric pressure | 101.325 kPa |
| Relative humidity | 40% |
| Mole fraction of oxygen in dry air | 0.20946 |
| Mole fraction of nitrogen in dry air | 0.790226 |
| Mole fraction of carbon dioxide in dry air | 0.000314 |
| TERRAIN CONSTANTS | |
| Pond water level | 8.0 m |
| Soil porosity | 55% |
| Volumetric water content at field capacity | 25% |
| Soil hydraulic conductivity, k | 1.11E-05 $m \cdot s^{-1}$ |
| Decomposing plant matter decay rate | 6.44E-08 $s^{-1}$ |
| Decomposing animal matter decay rate | 1.29E-07 $s^{-1}$ |
| Organic matter decay rate | 3.51E-08 $s^{-1}$ |
| Inorganic nitrogen decay rate | 9.51E-10 $s^{-1}$ |
| Seed decay rate | 3.22E-08 $s^{-1}$ |
| Soil nitrogen fixation rate | 9.49E-10 $kg \cdot s^{-1} \cdot m^{-2}$ |

Table 4.2: Parameter values used for the three baseline simulations.

| Description | Initial Value |
|---|---|
| SOIL INITIAL CONDITIONS | |
| Decomposing plant matter | 0.4 $kg \cdot m^{-2}$ |
| Decomposing animal matter | 0.1 $kg \cdot m^{-2}$ |
| Organic matter | 4 $kg \cdot m^{-2}$ |
| Inorganic nitrogen | 0.2 $kg \cdot m^{-2}$ |
| Seed size 1 | 0.01 $kg \cdot m^{-2}$ |
| Seed size 2 | 0.01 $kg \cdot m^{-2}$ |
| Seed size 3 | 0.05 $kg \cdot m^{-2}$ |
| Seed size 4 | 0.05 $kg \cdot m^{-2}$ |
| Seed size 5 | 0.1 $kg \cdot m^{-2}$ |
| Water table level | 8 m |
| MATERIAL STORAGE INITIAL CONDITIONS | |
| Mass of oxygen | 5E08 kg |
| Mass of nitrogen | 5E08 kg |
| Mass of water | 1E09 kg |
| Mass of carbon dioxide | 5E08 kg |

Table 4.3: Initial values for the encompassment and material storage realms in the three baseline simulations.

| Description | Initial Value | Initial Value |
|---|---|---|
| BIOLOGICAL COMPONENT INITIAL CONDITIONS | Baseline2 | Baseline3 |
| Species #5068: Coniferous tree | 0 instances | 400 instances |
| Species #5076: Deciduous tree | 0 instances | 400 instances |
| Species #5144: Perennial grass | 1500 instances | 1500 instances |
| Species #5146: Annual grass | 1500 instances | 1500 instances |
| Species #5156: Annual grass | 1500 instances | 1500 instances |
| Species #5166: Annual herb | 1500 instances | 1500 instances |
| Species #5320: Perennial grass | 1500 instances | 1500 instances |
| Species #5321: Nitrogen fixing perennial grass | 1500 instances | 1500 instances |

Table 4.4: Constitution and initialization of the biological component realms for the baseline simulations. (Baseline1 had no biological components.)

## 4.5. Results from the baseline simulations

Overall, the results from the experiments indicate that the model behaves as expected and that its performance is consistent. In each case, the total system mass remained relatively constant over the course of the simulation, with the maximum error in the mass balance being well within the range expected as a result of computer hardware precision limits (~10 kg in a total system mass of approximately 6.5E09 kg; all floating point variables used in the programs are 8-bit double precision). For all three simulations, approximately 62% of the total system mass was initially located in the encompassment and 38% was in material storage. In Baseline2 and Baseline3, the mass in the biological component realm was negligible compared to that of the other two realms. At the end of all the simulations about 64% of the system mass resided in the encompassment and 36% in material storage. Most of this shift occurred very early in the simulation due to a large transfer of water into the terrain as a result of rainfall.

The time histories of the total masses of the biological components, the material storage, the atmosphere, and the terrain for the Baseline1 experiment are plotted in Figure 4.9. Since there were no biological components present, the mass of this realm was zero throughout the simulation. After an initial transition period, the system shifted to fairly regular patterns of annual variation as it responded to the roughly cyclical forcing functions. For example, since the total pressure is controlled, the atmosphere's mass fluctuated according to an annual cycle, reflecting the yearly temperature pattern. Also, water was routinely transferred between storage and the terrain as part of the hydrologic cycle, resulting in minor mass variations in these areas.

Corresponding time histories for the Baseline2 and Baseline3 experiments are shown in Figure 4.10. From these, it is evident that the overall effect of biological components on the distribution of mass in the system (as compared to Baseline1) was a decrease in the material storage realm, coupled to a corresponding increase in the biological component realm. Also, there was a larger annual fluctuation of terrain mass, due to the yearly deposition of plant matter in the soil which subsequently decomposed. The biological component mass exhibited yearly fluctuations as well, with living plant biomass increasing over the summer growth period and then decreasing in the fall. In general, the shapes of the biological component curves are illustrative of the types of ecosystems specified. For Baseline2, the ecosystem is a grassland which reaches a fairly stable biomass level after about 30 years. For Baseline3, both herbaceous and tree species were included, with the former developing considerably faster. Thus, the ecosystem begins as a grassland and then, as the tree species become more established, undergoes a period of succession to forest. After about 50 years, as the trees begin to dominate, the total biomass starts to increase fairly sharply, and continues to increase as the forest becomes more established. This process was not completed after 75 years when the simulation was stopped.

The time histories of the total masses of the compounds in the ecosystem for the Baseline1 experiment are plotted in Figure 4.11 (dirt, $D_2$, is not shown since it

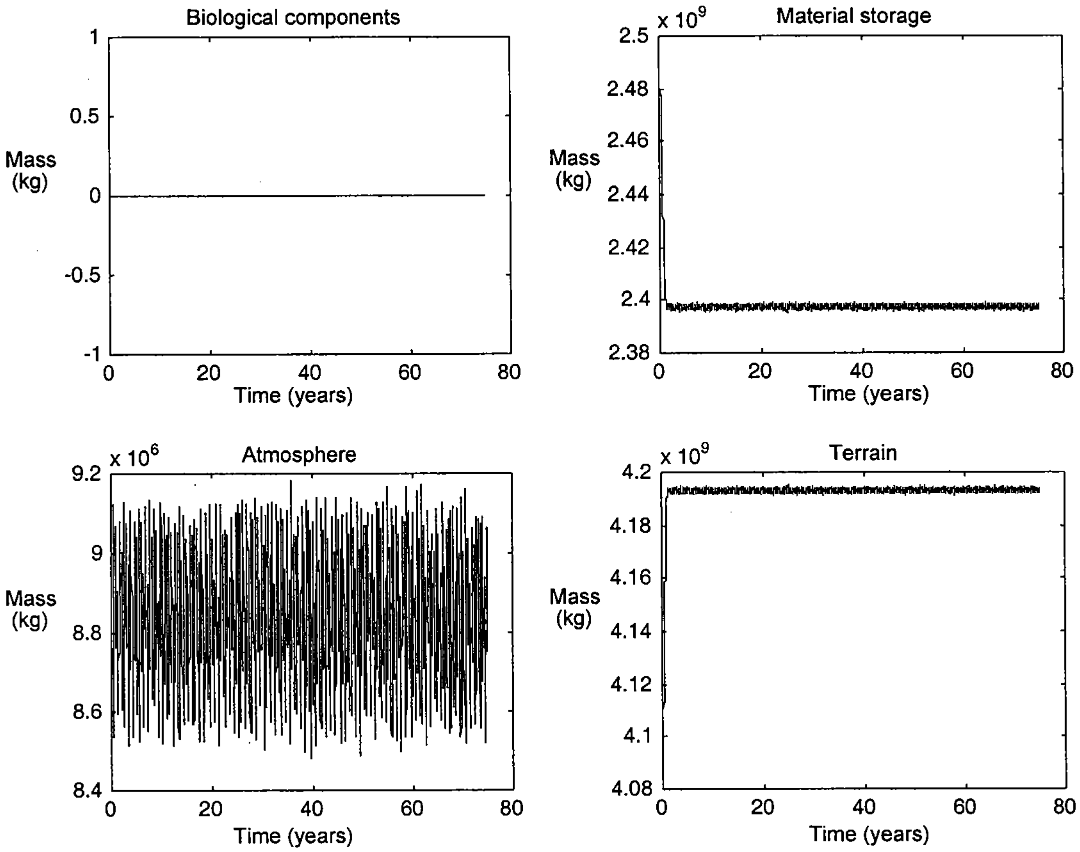


Figure 4.9: Total masses of the different ecosystem sections for the Baseline1 experiment. No biological components present.

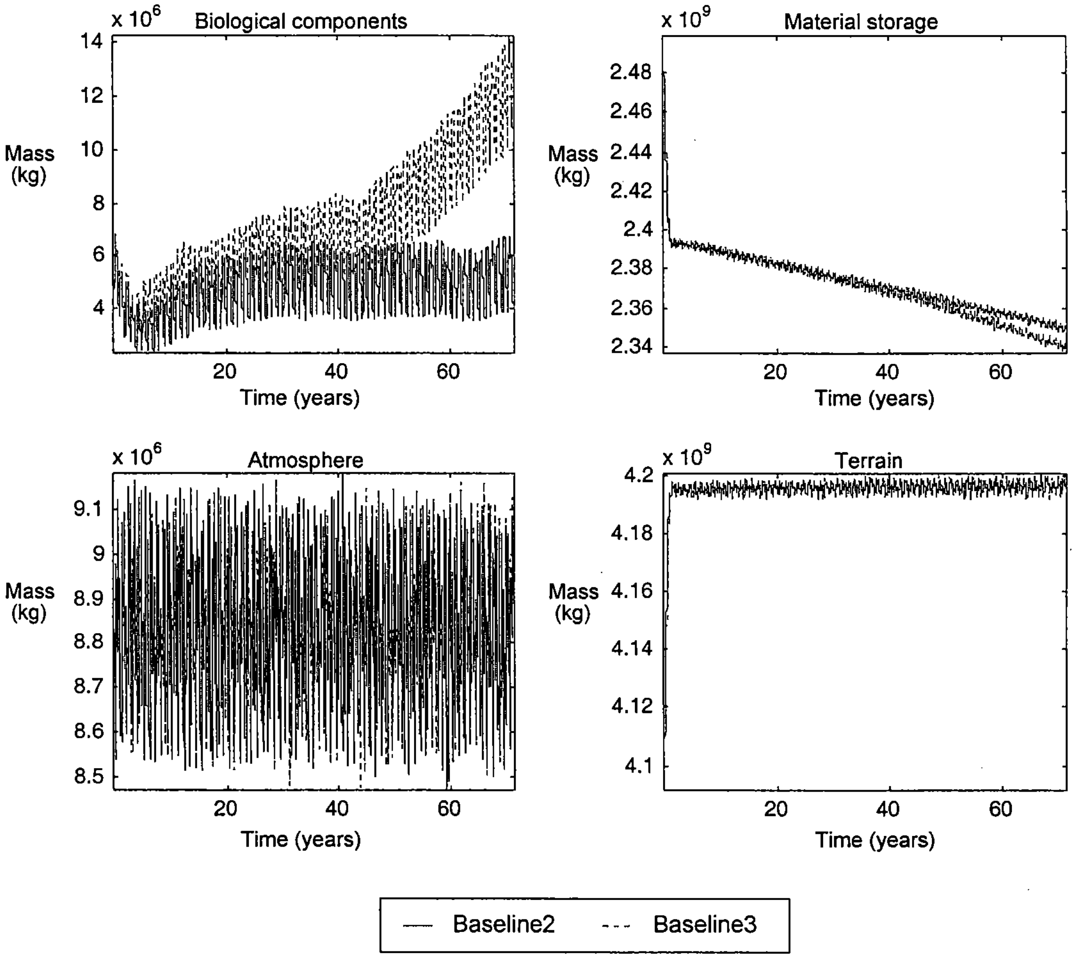


Figure 4.10: Total masses of the different ecosystem sections for the Baseline2 and Baseline3 experiments. Biological components: Baseline2, herbaceous species only; Baseline3, herbaceous and tree species.

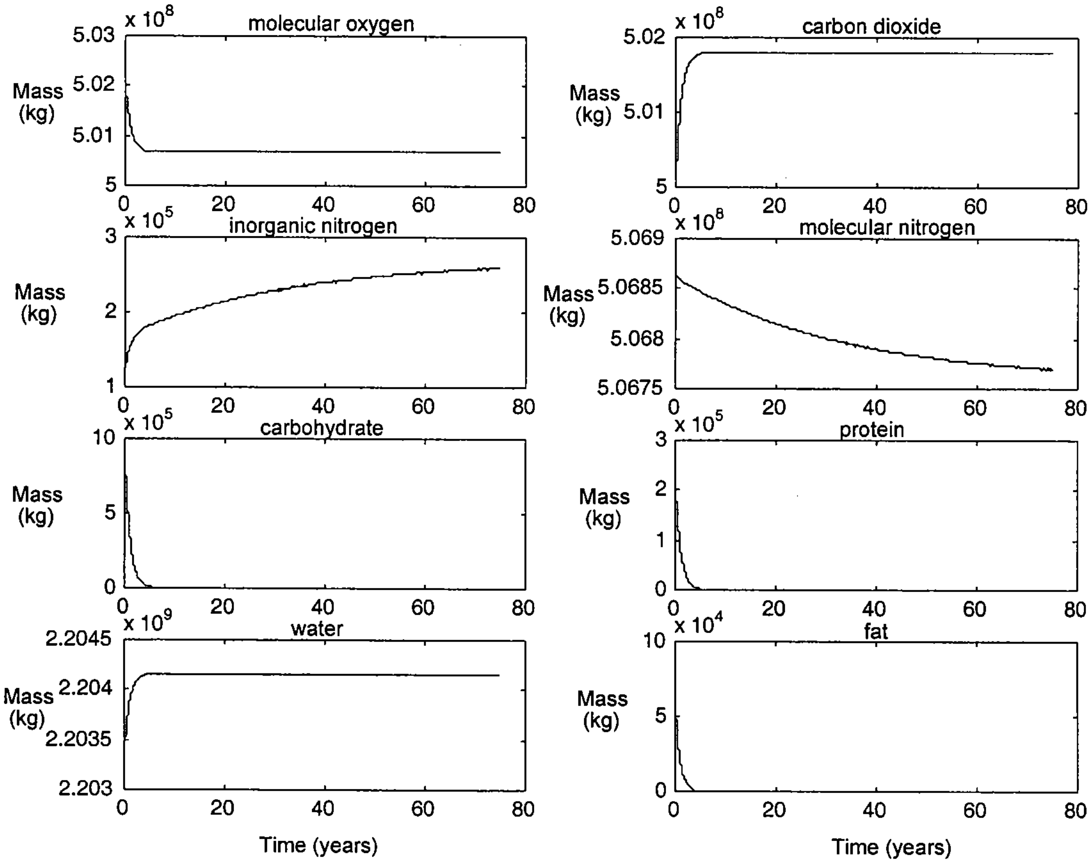


Figure 4.11: Total masses of the different compounds for the Baseline1 experiment. No biological components present.

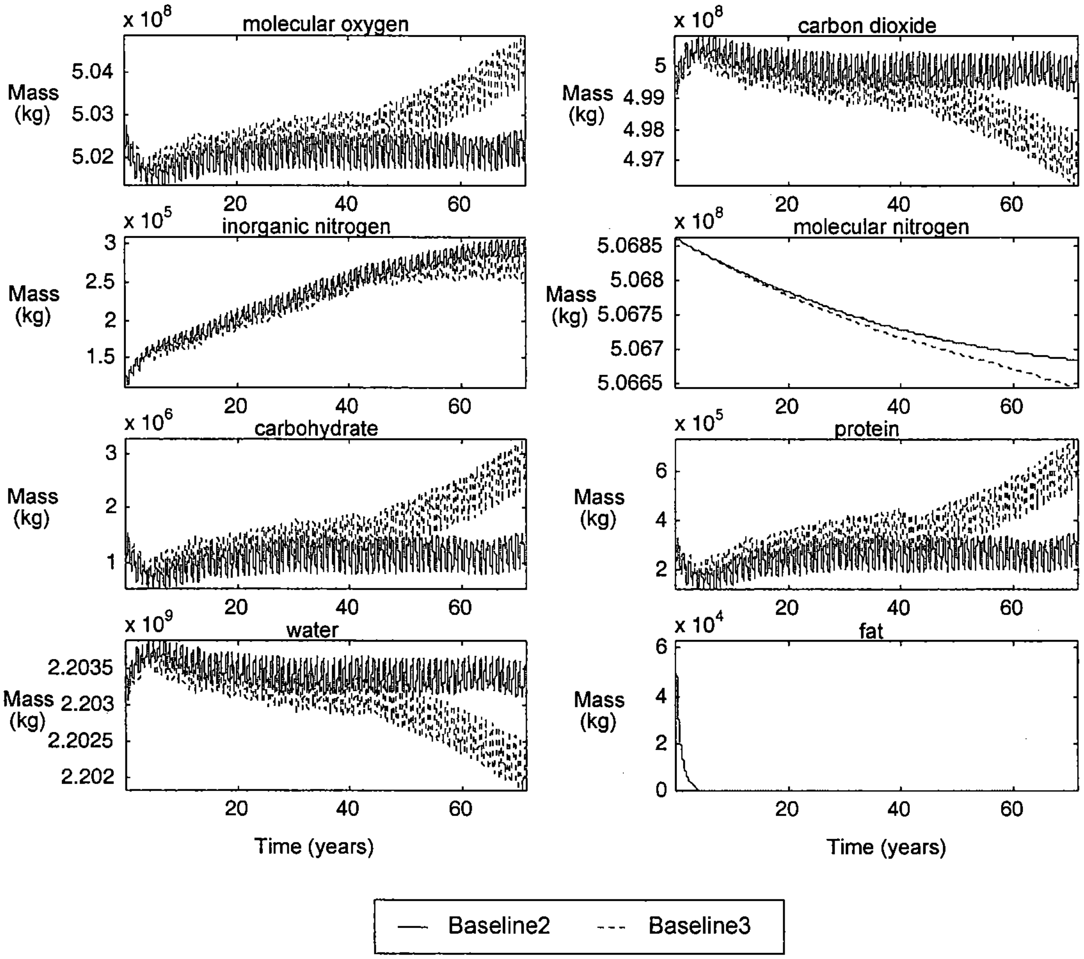


Figure 4.12: Total masses of the different compounds in the ecosystem for the Baseline2 and Baseline3 experiments. Biological components: Baseline2, herbaceous species only; Baseline3, herbaceous and tree species.

remained constant). With the exception of both molecular and inorganic nitrogen, all of the compounds reached a steady state fairly quickly (about 5 years), as would be expected for a system with no biological components. The soil, for instance, initially contains some organic and some decomposing material which were completely decomposed during the first few years. The quantities of the two nitrogen compounds continued to change due to the implicit soil nitrogen fixation process with which molecular nitrogen from the atmosphere is converted to inorganic nitrogen in the soil. These implicit biological processes in the soil were not "switched off" for the simulation.

The time histories of the total masses of compounds for the other two experiments are shown in Figure 4.12. The observable trends in these results are readily explained in terms of the plant life present. For instance, in both simulations, carbohydrate and protein mass, after initial decreases due to die-off, increased steadily. This is consistent with the overall increase in biological component mass that was noted. As would be expected, the general shapes of these curves are also similar to those obtained for the total mass of biological components (Figure 4.10). As well, both carbohydrate and protein, and related compounds such as molecular oxygen, carbon dioxide, water and inorganic nitrogen, all exhibited yearly fluctuations, in a manner consistent with the annual cycle of living plant biomass accumulation, and its subsequent deposition into the soil, and transformation via decomposition, etc. The total masses of carbon dioxide and water decreased with time as these compounds were broken down and incorporated into biomass. At the same time, the total mass of molecular oxygen increased because it is the product of photosynthesis. Fat, which was initially present in the soil as decomposing animal matter, disappeared altogether since it decays exponentially, and was not replenished in any way (plants do not contain fat). Lastly, molecular nitrogen decreased slightly due to nitrogen fixation which converted it into inorganic nitrogen that could then be absorbed by plants.

### 4.6. Conclusions

Overall, the results from the baseline simulations are promising; the trends in the distribution of mass between the different realms and sections thereof, as well as between the compounds, are readily explainable on the basis of the system configuration. In addition, both the configurations for Baseline2 and Baseline3 were viable; most of the species persisted for a reasonable length of time. The results from the three baseline simulations (and perhaps from other ones as well) will be used as standards to which other simulations will be compared, particularly those configured with more intricate food webs. Differences in the observed system comportment are expected to occur with the addition of animal species. For instance, carbon dioxide and oxygen production and consumption will likely be impacted as the total respiration by biological components is increased.

The use of a mass accounting approach in the model serves as a convenient check on program performance. During development, discrepancies in the mass balances were always directly traceable to logical or syntactical programming errors. Thus, when the various mass balances close for a simulation, and when the material distribution within the system makes sense in terms of expectations about a natural system, it inspires confidence in both the consistency of the model and the accuracy of the simulation that is based on it.

The model also shows promise with regards to its utility as a design tool. The differences in comportment between the grassland ecosystem model and the grassland-to-forest succession model were clearly evident, illustrating the model's potential in predicting both the qualitative and quantitative differences in overall system dynamics as a result of different initial configurations. In addition, what is perhaps the most interesting result is the fact that all of the observed system level trends arose from the specification and subsequent simulation of thousands of constituent components, none of which had any pre-programmed ability to direct the course of the system. Finally, the fact that the results are reasonable, i.e., they are not unlike what would be expected from a similar physical system, allows the conclusion to be drawn that an object-based approach is an applicable method by which to portray the dynamics of large scale ecological systems.

The model is relatively easy to configure and fun to use; it is always tempting to experiment further with related configurations, initial conditions, and forcing functions. In its overall representation of an ecosystem, the model meets the current objective of capturing the general dynamic patterns of a natural situation. It should, therefore, serve as a useful tool with which to test and develop complex control mechanisms for ecosystems as a means of exploring ecocyborg engineering in virtual space.

## 4.7. References

Beecham, J.A. and K.D. Farnsworth. 1998. Animal foraging from an individual perspective: an object orientated model. *Ecological Modelling* 113: 141-156.

Booth, G. 1997. Gecko: A continuous 2-D world for ecological modelling. *Artificial Life* 3(3): 147-163.

Clark, O.G., R. Kok, and P. Champigny. 1997. Generation of a virtual terrain. *Environmental Modelling and Software* 12(2/3): 143-149.

Dempster, W. 1993. Biosphere 2: System dynamics and observations during the initial two-year closure trial. SAE Technical Paper Series, #932290. SAE International, Warrendale, PA.

Fitz, H.C., E.B. DeBellevue, R. Costanza, et al. 1996. Development of a general ecosystem model for a range of scales and ecosystems. *Ecological Modelling* 88: 263-295.

Friend, A.D, A.K. Stevens, R.G. Knox and M.G.R. Cannell. 1997. A process-based, terrestrial biosphere model of ecosystem dynamics (Hybrid v3.0). *Ecological Modelling* 95: 249-287.

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.

Judson, O. 1994. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution* 9(1): 9-14.

Kawata, M. and Y. Toquenaga. 1994. From artificial individuals to global patterns. *Trends in Ecology and Evolution* 9(11): 417-421.

Kok, R., and R. Lacroix. 1993. An analytical framework for the design of autonomous, enclosed agroecosystems. *Agricultural Systems* 43:235-260.

Letcher, B. H., H. A. Priddy, J. R. Walters, and L. B. Crowder. 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.

Parrott, L., R. Kok and R. Lacroix. 1996. Daily average temperatures: Modeling and generation with a Fourier transform approach. *Transactions of the ASAE* 39(5): 1911-1922.

Parrott, L. and R. Kok. 2000a. A generic primary producer model for use in ecosystem simulations. Chapter 5 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000b. Higher trophic levels in a generally configurable ecosystem model. Chapter 6 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000c. Implementation of a generally configurable object-based ecosystem model. Chapter 3 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000d. Incorporating complexity in ecosystem modelling. Chapter 2 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Taylor, C., D. Jefferson, S. Turner and S. Goldman. 1988. RAM: Artificial life for the exploration of complex biological systems. In *Artificial Life,* ed. C. Langton, 276-295. Reading, MA: Addison-Wesley.

Wolff, W. F. 1994. An individual-oriented model of a wading bird nesting colony. *Ecological Modelling* 72: 75-114.

## 4.8. Appendix I

### *4.8.1. Nitrogen fixation by soil micro-organisms*

The number of kilograms of nitrogen fixed per time step per grid cell is calculated as:

$$N_{fixed} = \begin{cases} 0 & T_a < T^* \\ Nfixrate_{20}(\Delta t)(L)(W)\dfrac{(T_a - T^*)}{(20 - T^*)} & T_a \geq T^* \end{cases}$$

where:

$N_{fixed}$ = amount of inorganic nitrogen fixed (kg)

$Nfixrate_{20}$ = "nitrogen fixation rate" at 20°C ($kg \cdot s^{-1} \cdot m^{-2}$)

$\Delta t$ = time step (s)

L = cell length (m)

W = cell width (m)

$T^*$ = minimum temperature at which N-fixation occurs (°C)

$T_a$ = air temperature (°C)

### *4.8.2. Saturated water flow in the soil*

The amount of saturated water that flows from one grid cell to adjacent cells during a given time step is calculated as follows:

$$A_{12} = kL\Delta t(h_1 - h_2)\gamma_w$$
$$A_{13} = kW\Delta t(h_1 - h_3)\gamma_w$$

where:

A = amount of water to transfer between two cells (kg)
k = soil hydraulic conductivity ($m \cdot s^{-1}$)
L = cell length (m)
W = cell width (m)
$\Delta t$ = time step (s)
h = saturated water table height from bottom of soil surface (m)
$\gamma_w$ = specific gravity of water ($kg \cdot m^{-3}$)

This calculation is performed in turn for each cell on the terrain. A cell's "neighbours" are the cell directly to one side (right) and the cell directly above it. Wrapping of the landscape occurs in one direction on the grid (see Figure 4.5, terrain topography).

## 4.9. Appendix II - Simulation Time Trials: Assessment of the model's sensitivity to changes in the time increment.

### *4.9.1. Method*

A number of simulation trials were run to assess the sensitivity of the ecosystem model to changes in the time increment, $\Delta t$. The trials were initialized with identical starting states (i.e., the same set of input files was used for each of the experiments), except for the value of $\Delta t$, which was varied between 60 and 3600 seconds for the different simulations. Each trial was run for one simulated year, or until an error occurred, in which case the selected time increment was assumed to be inappropriate. For the trials that successfully ran for one year, the final population sizes and the masses of each of the ecosystem realms were compared with a control trial ($\Delta t$=600 s) and any differences were noted. The final states of individual components were not inspected for coherency.

For all trials, the encompassment and material storage realms were configured as per the baseline simulations (Tables 4.2 and 4.3), and the biological component realm was configured with herbivores, herbs, and trees. The same forcing functions were used for each trial, and the random number generator (used in some biological component methods to make probabilistic decisions, and used to generate attribute values for new instances) was initialized with the same seed value so as to produce the same series of random numbers for each case.

### *4.9.2. Results*

The initial conditions and ending states for the time trials are shown in Table 4.5 (following page).

| Initial Population Size (# of instances)*** | | | | | | | | | Initial Masses of Ecosystem Realms (kg) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Encompassment | Material Storage | Bio. Comp. |
| 5 | 10 | 10 | 10 | 100 | 100 | 100 | 100 | 100 | 4099436249 | 2500000000 | 7597 |

| | Δt (s) | Total Runtime** | Final Population Size (# of instances) | | | | | | | | | Final Masses of Ecosystem Realms (kg) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Encompassment | Material Storage | Bio. Comp. |
| Trial 1* | 600 | 2 h | 1 | 10 | 11 | 30 | 131 | 192 | 81 | 70 | 98 | 4196977493 | 2402454887 | 11466 |
| Trial 2 | 60 | 22 h | 1 | 10 | 9 | 29 | 133 | 342 | 41 | 11 | 47 | 4196019817 | 2403410019 | 14011 |
| Trial 3 | 3600 | < 1 h | ERROR: Halted at 344 days due to system mass being substantially different from start mass. | | | | | | | | | | | |
| Trial 4 | 1800 | < 10 min | ERROR: Halted at day 3 due to incoherent plant object (negative carbohydrate mass in leaf). | | | | | | | | | | | |
| Trial 5 | 1200 | 1 h | 1 | 10 | 11 | 30 | 133 | 189 | 77 | 69 | 103 | 4191490411 | 2407941916 | 11520 |
| Trial 6 | 1500 | 50 min | 1 | 10 | 9 | 29 | 136 | 399 | 73 | 103 | 106 | 4197286474 | 2402142674 | 14698 |
| Trial 7 | 1750 | 45 min | 1 | 10 | 10 | 30 | 137 | 204 | 78 | 70 | 101 | 4196551572 | 2402880686 | 11589 |

* Control experiment

** Hardware: PentiumPro 200Mhz, 128Mb RAM; Operating system: Windows 98 (Runtime is reported in real hours.)

*** Selected species:

| **No.** | **ID** | **Description** |
|---|---|---|
| 1 | 5001 | small herbivore |
| 2 | 5068 | tree |
| 3 | 5076 | tree |
| 4 | 5087 | herb |
| 5 | 5144 | herb |
| 6 | 5146 | herb |
| 7 | 5156 | herb |
| 8 | 5166 | herb |
| 9 | 5320 | herb |

Table 4.5: Initial conditions and ending states for ecosystem simulation time trials

### 4.9.3. Discussion

It appears that the largest Δt that can be used without obtaining erratic results is somewhere between 1750-1800 seconds, which is about 30 minutes. The results based on this time increment are quite different from those obtained for the control trial, the most notable discrepancy being the final population sizes of the different species. This was the major difference observed for all of the trials that ran successfully. This effect may be due to the random number generator being called less frequently (or more frequently in the case of smaller time increments), so that the organisms are acting according to different random inputs. The magnitude of the time increment will also affect the functioning of certain methods such as photosynthesis, growth, etc. The final masses of the ecosystem realms were also somewhat different for the five successful trials, reflecting the different numbers of biological components.

Although these tests seem to indicate that plant-based ecosystems, and perhaps other systems with more elaborate biological component configurations, could be run with a time increment that is somewhat larger than the value currently used, it is not clear what effect this would have on animal behaviour, since only a few instances of one animal species were included in these trials. For animals, social and consumer-based behaviours are modelled that, in physical reality, often occur at a much smaller time scale than even 10 minutes. Thus, the use of a larger time increment would severely limit the types of activities that could be modelled (particularly in the future, when animal behaviour may be represented in much greater detail than it is in the current model) and/or affect the manner in which these activities could be represented. For these reasons, given the type of animal behaviour that is being modelled, and the degree of detail that is included in all aspects of the model, it was concluded that smaller time increments are best, even at the cost of longer simulation runtimes.

## *Connecting text between Chapters 4 and 5*

In Chapter 4, a general overview of the ecosystem model was presented. In Chapter 5, further details regarding the representation of plant life are provided. Simulation results are discussed with reference to the overall behaviour and population dynamics of plants in the ecosystem, and the response of plants to predation is studied.

The plant methods discussed in Chapter 5 are encoded in the file "bio.c", available for viewing on the enclosed CD-ROM (Appendix B).

Chapter 5 will serve as the working text of a paper being coauthored by L. Parrott and R. Kok.

# Chapter 5: A generic primary producer model for use in ecosystem simulations.

## Abstract

This article is a description of a generic primary producer (plant) model that has been developed as part of a larger whole ecosystem model project. The plant model is generic in the sense that it can be configured to represent many different types of plants, including herbs, bushes and trees. The model is object-based, meaning that each plant, or small collection of plants, is represented by a unique "object" with attributes such as mass and photosynthesis rate. Population level dynamics are therefore governed by object-level rules that describe processes such as growth and reproduction. The plant objects are distributed over a spatially explicit terrain, and their behaviour is driven by irregular climatic forcing functions. The plant canopy is modelled with three layers of vertical differentiation, which affects inter-species competition for available radiant energy. In addition to competing with other plants for essential nutrients, sunlight and water, the plant objects may also be subject to grazing by animals. Results of initial simulations with small collections of species distributed about a 250,000m$^2$ terrain are presented.

## 5.1. Introduction

We are studying how to engineer controlled, materially closed ecosystems, similar to those that might be installed in extra-terrestrial space stations. This work is part of a larger research initiative called the EcoCyborg Project, for which the overall objective is to learn how to engineer all types of sparsely connected autopoietic networks, or *biosystems* (Clark, 1999). Our present approach is to develop and use models and simulations with which to test different biosystems engineering strategies, as a means of exploring the relationships between the constitution and resultant comportment of such systems. Our specific goal is to develop a computational model of a materially closed ecosystem that has been enhanced with technological control components: an *ecocyborg*. A set of interlinked models is being written to represent the various portions of an ecocyborg, one being a generally configurable ecosystem model. In this article, an overview of the approach adopted to represent primary producers (plants) within the ecosystem model is provided.

In the ecosystem model, fairly complex food webs (up to 1000 species) can be accommodated, including primary producers, as well as consumer organisms at higher trophic levels. The various organisms interact within a spatially explicit, temporally variable environment subject to uncontrolled forcing functions (weather). Since a key aspect of the ecocyborg is material closure, the cycling of mass between all of the biological components and their environment is a fundamental feature of the ecosystem model, with every mass exchange being completely accounted. The model is time driven; a 10 minute time increment is being used for current simulations.

The primary producers are represented using an approach that captures the basic functions of higher plants to a reasonable, but not exhaustive, degree of detail. A main goal in developing the plant model was that it be sufficiently generic to be applicable to wide range of vegetation types, including herbs, bushes and trees. As a result, it does not provide an accurate representation of any particular plant species. When used to simulate the behaviours of a large number of plants belonging to different species, however, it does provide reasonable predictions of total biomass accumulation in an ecosystem, in addition to depicting the major influences of plants on the soil and atmospheric environments.

Thus, for this part of the project, the objectives have been to create a model of primary producers that:
(1) fits within the overall framework of a larger, object-based ecosystem model, and therefore meets certain constraints with regards to mass accounting schemes, object interactions, etc.
(2) is sufficiently flexible and generic so as to be applicable to many kinds of plants (e.g., herbs, bushes and trees), simply by changing the values of a few parameters; and,
(3) is based on processes that are driven, directly or indirectly, by climate related

conditions.

The plant model that has been developed is described in detail in the sections that follow, together with sample results to illustrate its behaviour in simulation.

### 5.2. Approaches to plant modelling

Plants can be modelled at different resolutions, or levels of description, depending upon the objectives of the endeavour. For example, an agronomist desiring to predict yields might represent soybeans using a crop level model, a silviculturalist studying the effects of fire on a forest stand might model trees as individual plants, and a biologist studying photosynthesis might model the biochemical processes occurring in a leaf at the cell, organelle or molecular level. In ecology, plants are usually modelled as part of an ecosystem, and in this context, they are most often represented at the crop or whole plant level, since the aspects of interest are usually those that involve interactions between the organisms and their environment. In many of the global climate models, for example, vegetation is represented as large masses of forest stands or grasslands (e.g., Fitz et al., 1996). In models with which food web interactions or spatial variation are studied, plants are more likely to be represented as individual units.

A number of authors have discussed the advantages of individual-based, whole plant models in ecology in lieu of crop, or stand level approaches (Hogweg and Hesper, 1990; Clark, 1992; Huston, 1992). In such models, each plant is represented separately, allowing for the consideration of the effects of individual plants on the local microenvironment, which in turn influences the structure and composition of much larger regions. When implemented in simulation, however, individual-based models can quickly become very computationally intensive. They are therefore most often used to represent large plants such as trees on small sites of 0.1 to 10 ha. The GAP models, for example, are a widely used class of individual-based tree models used to simulate the structure of forest stands in small plots (Huston, 1992; Shugart et al., 1992). In these, the plot size is specified such that it is equal to the zone of influence of a plant of maximum size.

Thus, a number of plants of all sizes will compete for dominance on a plot, and the death of a large individual can dramatically change the light environment lower in the canopy. Although individual-based plant models are usually employed to represent the dynamics of populations on fairly small sites, it can, however, be argued that they are also appropriate for the representation of much larger systems. Shugart et al. (1992), for example, pointed out that simulations based on such models are highly applicable to the prediction of the effects of global climate change, since these models can "change composition in response to physical disturbance and environmental changes".

A few individual-based plant models have been written to represent large spatial areas in order to study the effects of phenomena such as global climate change on vegetation. With the Hybrid 3.0 model (Friend et al., 1997) individual trees and a grass layer are modelled on small plots which can then be linked together to represent larger areas. The behaviour of trees is simulated on an annual time step, and the grass is simulated on a daily time step. The soil-plant-atmosphere system is completely coupled, and a mass balance approach is used to predict and track material exchanges. Plant growth is driven by climatic conditions, and a vertically differentiated canopy is used to estimate the incident irradiance received by foliage at different layers. In the TROLL model (Chave, 1999), a similar approach is used to represent small plots of tropical rainforest. In TROLL, however, only trees are represented. Each tree is modelled individually, and simulations are executed on a one year time step. Simulations based on TROLL have been executed for 20km$^2$ plots with 20 million trees on a Cray computer. EFIMOD (Chertov et al., 1999) is another forest growth model in which, like Hybrid 3.0, trees are modelled individually and understory vegetation is modelled as a single mass below. EFIMOD is also executed with an annual time step and has been used to study the effects of climate change and pollution on European forests for 100 year periods.

Regardless of the level of resolution at which they are developed, plant models may be of three general types: empirical, mechanistic, or teleonomic (Thornley and Johnson, 1990). Empirical plant models are direct representations of observed data. For example, the stomatal conductance of tomato plants under a variety of different temperature conditions could be measured and recorded, and a curve could be fit to the data. The equation of the curve would then be an empirical model of that particular relationship. Evidently, such models are generally not based on more abstract scientific principles or knowledge, nor is there usually any consideration, or explicit representation, of the underlying mechanisms that give rise to the phenomenon that is modelled. They are, however, quite accurate descriptions of the data sets they represent, and there have been many useful empirical models developed at both the crop and whole plant level (see references in Thornley and Johnson, 1990). The drawback of an empirical model is that it is usually only applicable to the particular species upon which it was based, and is not easily adapted to other types of plants.

In contrast, mechanistic models are based on representations of the underlying mechanisms in a plant that contribute to its growth and functioning. These models tend to be more broadly applicable to different types of plants and are often written to provide some insight into the nature of plant physiology. "Transport-resistance" models (Thornley and Johnson, 1990; Thornley, 1976; Thornley, 1991), in which material resources are allocated throughout a plant according to concentration gradients, are a common application of the mechanistic approach. In this way, the overall growth of the plant arises in an undirected manner from the specification of lower level processes.

In teleonomic models, a supervisory principle, or "invisible hand" is assumed to direct the functioning and development of a plant. The most common application of this approach is in resource partitioning models in which the plant is assumed to "want" to maintain a certain shoot:root ratio, and new growth is apportioned according to this overriding rule (Thornley and Johnson, 1990). Usually, the supervisory principle that is included in a teleonomic model will have been derived from empirical observations. As in empirical modelling, although no adequate theoretical framework is available with which to explain the relationship, the rule will reflect an observable and "real" feature of a system. Thus, often the teleonomic approach is the most practical one when limited mechanistic explanation is possible, but a more complete understanding of the system is wanting. It is also a useful approach when a detailed mechanistic model would be too complicated to implement. In this case, it may be possible to summarize a substantial fraction of the total set of mechanisms with one or more guiding principles.

The model that we have developed for use in ecosystem simulations is a hybrid of the approaches described above. First, the model represents the state and behaviour of a plant *object*, which may be a lump of small plants or a single plant, depending upon the species. Thus, like the EFIMOD and Hybrid 3.0 models, trees are modelled on an individual basis, and understory vegetation (herbs and grasses) are modelled as lumps. The model is, therefore, a hybrid of individual and crop level approaches. The state of each plant object is stored as a collection of over 100 attributes (e.g., mass, photosynthesis rate, leaf area, etc.) whose values are modified by functions such as photosynthesis, respiration and water absorption that together constitute larger processes such as growth. Most of the functions are based on a combination of mechanistic and empirical relationships, and are driven by external climatic conditions as well as internal conditions related to the state of the plant. A few functions, such as those related to the translocation of photosynthates to the various parts of the plant, are teleonomic. Overall, the behaviour of the plants results entirely from the execution of functions by plant objects (this is also true for objects that represent higher trophic level organisms in the ecosystem model). The overall dynamics of the ecosystem is therefore governed by, and emerges from, object-level rules.

This model is different from similar forest or ecosystem models in a number of ways. First, the plant objects are completely configurable, so that a wide variety of different generalised plant types (e.g., deciduous tree, evergreen tree, annual grass, etc.) can be represented. Second, the behaviour of the plants is simulated at a much shorter time step than that which is usually used for vegetation models, which allows for a much more detailed representation of plant processes. Thirdly, animals are included that consume the plants. This allows for the effects of predation on plant growth and spatial distribution patterns to be studied. These effects, which can be substantial, are not considered in most large vegetation models.

## 5.3. Description of the plant model

### *5.3.1.Overview*

Within an ecosystem, plants play an important role as primary producers, and, accordingly, as an essential component of the overall food web. In developing an appropriate plant model, therefore, our objective was to create a model of plant growth and development that was sufficiently generic to be applicable to all types of plants (e.g., herbs, bushes and trees), yet was detailed enough to allow for a reasonable representation of material flux and accumulation within the vegetative zone. Since we are more interested in studying the global-level behaviour of ecosystems, an emphasis was placed on modelling functions such as photosynthesis and respiration, in which matter is exchanged between the plant and its environment, rather than on the internal physiological aspects of plant growth and development. Additionally, competition between plants for resources such as energy and water is an important contributor to the formation and structural organization of plant communities and was therefore considered to be an essential aspect of the plant model.

As previously mentioned, the plant model was developed to fit within the framework of an overall ecosystem model. The latter represents a materially closed system, thus, a complete mass accounting is done for each of the components in the ecosystem, as well as for the system as a whole. So as to keep the model and the mass accounting procedure relatively simple, everything in the ecosystem is assumed to be composed of one or more of nine different compounds, each of which is made up of one or more of five elements (Table 5.1). (The compounds and elements correspond fairly closely to those found in nature, although a fictitious "dirt" element (D) and a related compound ($D_2$) were used to represent the bulk of the soil that is normally relatively inert in an ecosystem.) Correspondingly, modelled plants are presumed to be composed primarily of water, with some carbohydrate and protein mass. They also contain a small reserve of inorganic nitrogen absorbed from the soil for use in protein synthesis. The soil in which the plants grow is represented as being made up of inorganics ($D_2$ & $N_2$), as well as decomposing organic material (in turn made up of fat, carbohydrate, protein and water) deposited by plants and animals. Organic compounds in the soil are decomposed into atmospheric $H_2O$, $N_2$ and $CO_2$ as well as inorganic nitrogen that remains in the ground and is later absorbed by growing plants. The soil pores may be filled to various degrees with water that is available for uptake by the plants.

| **The Elements** | | **The Compounds** | |
|---|---|---|---|
| *Name* | *Symbol* | *Name* | *Composition* |
| Carbon | C | Carbohydrate | $C_6H_{12}O_6$ |
| Dirt | D | Carbon Dioxide | $CO_2$ |
| Hydrogen | H | Compound Dirt | $D_2$ |
| Nitrogen | N | Fat | $C_{57}H_{110}O_6$ |
| Oxygen | O | Inorganic Nitrogen | $N_2$ |
| | | Molecular Nitrogen | $N_2$ |
| | | Molecular Oxygen | $O_2$ |
| | | Protein | $C_4H_{12}O_4N_2$ |
| | | Water | $H_2O$ |

Table 5.1: Compounds and elements used to make up materials in the modelled ecosystem.

The ecosystem model is spatially explicit. A modelled plant is therefore always situated within one of 2500, 10m x 10m rectangular grid cells into which the terrain is divided (other terrain sizes and grid cell resolutions can be specified but are not used for the simulations described here). Each plant absorbs water and nutrients from the terrain grid cell in which it is located, and captures radiant energy in order to fuel its growth. Between-plant competition for these resources is discussed in further detail below. Soil depth and rainfall rate may vary between grid cells, and saturated water flow between cells occurs along the hydraulic gradients that arise as a result of rainfall, or other water transfer activities such as absorption by plants. Incident irradiance at the top of the canopy and ambient air temperature is assumed to be uniform across the terrain. Radiation, temperature and rainfall values are generated by an external weather model and serve as ecosystem forcing functions.

Herbs and grasses are treated as "lumps" of vegetation (divided up into "single plant" units of 1 square decimetre, covering a maximum area equal to that of one grid cell) and all other types of plants are modelled as individual organisms. Each lump or individual plant is represented by a distinct object (also called a species "instance"), which is assumed to be composed of six parts: seed, organic storage, leaf, stem, root and inorganic storage. Each of these is, in turn, composed of water, carbohydrate and protein (except for the inorganic storage which is entirely made up of inorganic nitrogen). The "optimal" fractions of water, carbohydrate and protein in a plant are species level attributes, the values of which are constant throughout the life of the plant. Except for inorganic storage, there is no differentiation of plant parts on this basis: In a healthy plant, each part is made up of the same optimal fractions of water, carbohydrate, and protein. Despite the fact that they all have the same material composition, division of the plant into these six parts serves a number of purposes. For example, it allows for a more accurate prediction of photosynthesis rate (which is a function of total leaf area) as well as water and nutrient usage. In addition, it provides the possibility for the adoption of selective feeding behaviour by consumers (e.g., an insect that consumes only the stem and leaves, or a bird that eats seeds) and the representation of the effects of this on plant health and development. Thus, this approach provides an effective compromise between the requirement for adequate detail in the plant model, and the need to take practical limitations into consideration.

In the ecosystem model, descriptions of up to 1000 species are accommodated, any number of which may be plant species. Each of these is represented by a corresponding population of plants, which are represented by objects. A plant object has a number of attributes that describe its state, and a number of methods, or functions, that describe its potential behaviour. All plant objects of the same species have the same attributes and methods, although the values of some attributes (e.g., mass) may vary amongst plants in a population. Thus, each time a new plant is created, it is initialised with a number of object-specific attribute values which are selected stochastically (from within a pre-specified range that describes the ideal distribution of that attribute value amongst plants in a population), and with species-specific attribute values that are the same for all objects of its species. While object-specific attribute values serve to describe the current state of a plant and are variable, species-specific attribute values serve as plant model parameters that define each generalised plant type, or species. In this manner, both within and between species variation in the plant community is accommodated. (See Parrott and Kok, 2000a, for a detailed description of the implementation of objects in the ecosystem model.) A list of all attributes used to describe the state of a plant object is given in the article appendix.

The overall functioning of a modelled plant of any species is illustrated in Figure 5.1. In general, a plant first photosynthesizes, creating new carbohydrate assimilate. Some of this assimilate is then used to meet the respiration requirements of the plant. A portion of the remaining carbohydrate assimilate is made into protein which is subsequently distributed along with the carbohydrate to the various plant parts according to a teleonomic, "source-sink" allocation scheme. Other ongoing processes include nitrogen and water absorption and litter production. The way in which all of these functions are modelled is described in further detail below.

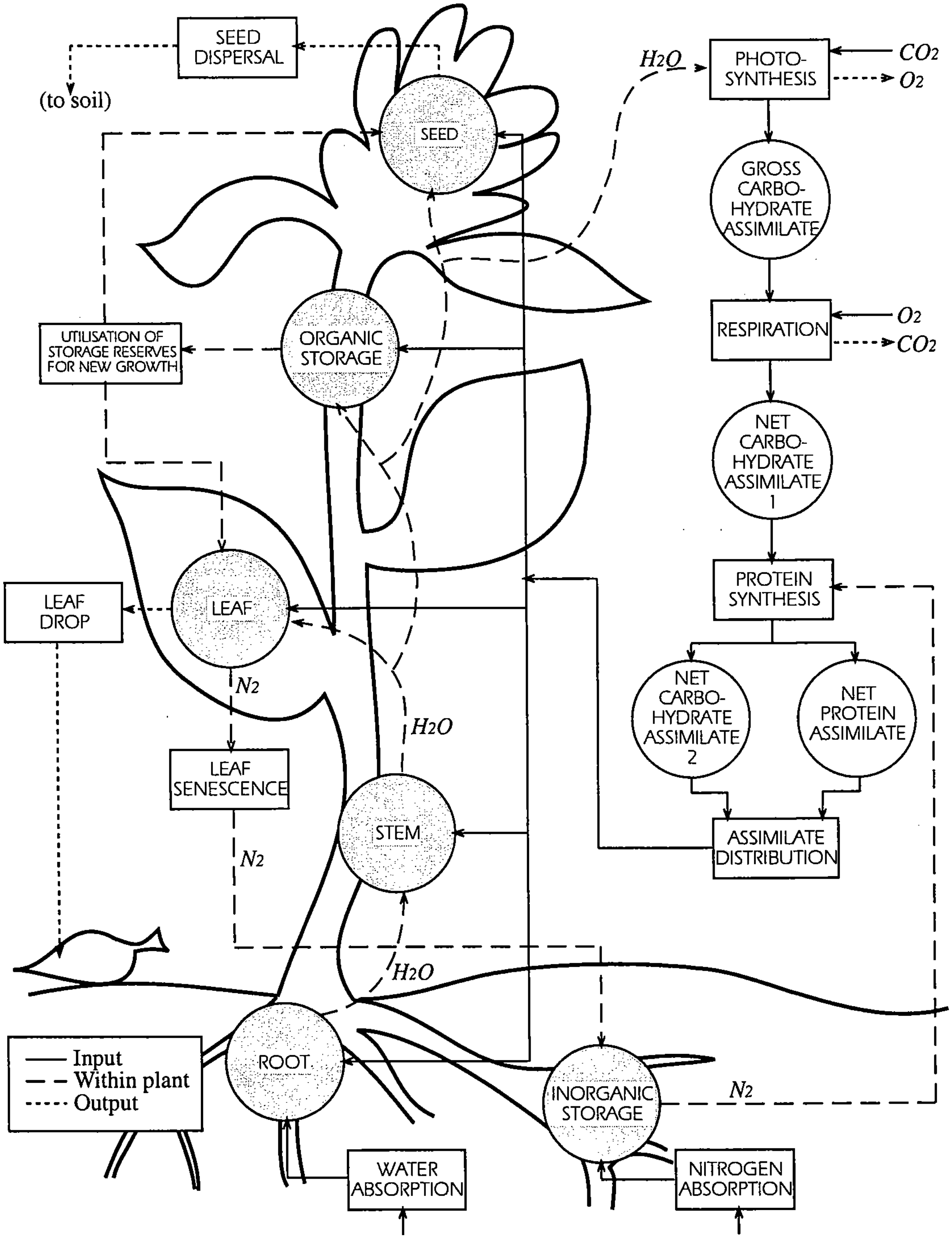


Figure 5.1: Parts, main processes and material flows in a generic plant object.

### *5.3.2. Photosynthesis*

The single largest factors affecting plant development are radiant energy absorption and its subsequent utilization for the production of simple sugars.

The general chemical process is:

$$6CO_2 + 12H_2O + \text{radiant energy} \dashrightarrow 6CH_2O + 6H_2O + 6O_2 \tag{1}$$

The kinetics of the process are modelled using a non-rectangular hyperbola (modified Michaelis-Menten equation) to describe the rate of leaf photosynthesis as a function of incident irradiance. The equation used is that given by Thornley and Johnson (1990, p. 228 Eqn. 9.10i):

$$P = \frac{1}{2\theta}\left\{\alpha I_l + P_m - \left[(\alpha I_l + P_m)^2 - 4\theta\alpha I_l P_m\right]^{1/2}\right\} \tag{2}$$

where:

$P$ = gross leaf photosynthesis, $kg_{CO_2}\cdot m_{leaf}^{-2}\cdot s^{-1}$

$\theta$ = empirical parameter, $0 < \theta < 1$

$P_m$ = maximum gross photosynthesis rate at light saturation, $kg_{CO_2}\cdot m_{leaf}^{-2}\cdot s^{-1}$

$\alpha$ = photosynthetic efficiency of the leaf, $kg_{CO_2}\cdot J^{-1}$

$I_l$ = irradiance incident on the leaf surface, $W\cdot m_{leaf}^{-2}$ of PAR (photosynthetically active radiation). $I_A$, $I_B$ or $I_C$ is used, see Equations 8-10.

This equation has been fit to data from a wide range of crop plants (Marshall and Biscoe, 1980; Acock et al., 1978), and appears to provide a good estimate of gross leaf photosynthesis under ordinary growing conditions. However, neither changes in temperature, nor atmospheric $CO_2$ concentration, nor leaf nitrogen content are taken into account, although all of these may affect a leaf's ability to photosynthesize and, therefore, the values of $P_m$, $\alpha$ and $\theta$ (Friend, 1995; Smith et al., 1993; Thornley, 1998).

The atmosphere in the modelled ecosystem is controlled (pressure: 101.325 kPa, relative humidity: 40%, $CO_2$ mole fraction of dry air: 0.000314) such that the gas phase concentration of $CO_2$ will vary slightly due to temperature fluctuations and the partial pressure will remain stable. The effects of $CO_2$ concentration on the plants was therefore not considered when calculating photosynthesis rates. On the other hand, marked fluctuations in air temperature and leaf nitrogen content do occur in the modelled ecosystem and are accounted for with the photosynthesis model through modifications to the value of $P_m$.

Although changes in ambient temperature have been shown to affect all three of the parameters used in Equation (2) ($P_m$, $\alpha$, and $\theta$), only $P_m$ is significantly affected (Thornley and Johnson, 1990). Over the normal range of growing temperatures (e.g., 5-25°C), $P_m$ has been observed to vary approximately linearly with temperature. Thus, this relationship is incorporated into the photosynthesis model by using the following equation (Thornley and Johnson, 1990, p. 229) to select a value for $P_m$:

$$P_m = \begin{cases} 0 & T < T^* \\ P_m(20)\left(\dfrac{T - T^*}{20 - T^*}\right) & T \geq T^* \end{cases} \tag{3}$$

where $P_m(20)$ is the value of $P_m$ at 20°C and $T^*$ is the temperature at which photosynthetic activity ceases (this is a species-specific attribute).

Leaf nitrogen content has also been shown to significantly affect the photosynthetic rate in physical plants according to a roughly linear relationship

(Thornley, 1998). In the modelled plants, nitrogen is present in their protein and also in the inorganic storage. Although the storage reserve may vary as it is used for new growth, leaf protein content is maintained at a constant value, except during the end of the growing season in perennial herbs and deciduous plants. During this phase, leaf protein is broken down (see Section 5.3.10, below) and the nitrogen is transferred to storage. To provide a rough representation of the effect of this on leaf photosynthesis, the value of P, as predicted by Equation (2) is assumed to decrease linearly with decreasing leaf protein concentrations according to the following relationship:

$$P_{final} = P * s \tag{4}$$

where,

$$s = \frac{m_{leaf.prot} / m_{leaf}}{opt.prot} \qquad 0<s<1 \tag{5}$$

and,

$m_{leaf.prot}$ = mass of protein in plant object's leaf part, kg

$m_{leaf}$ = mass of plant object's leaf part, kg

$opt.prot$ = "optimum" fraction of protein in leaf

The total amount of $CO_2$ used in the production of photosynthates by a plant over a given time interval is, therefore, calculated as:

$$P_c = P_{final} * \Delta t * A_{leaf} \tag{6}$$

where,

$P_c$ = mass of $CO_2$ used by the plant in photosynthesis, kg

$\Delta t$ = simulation time increment, s

$A_{leaf}$ = total area of leaf on plant, $m^2$ (plant leaf mass/leaf thickness)

Once $P_c$ is known, the mass of carbohydrate assimilate produced by the plant as a result of photosynthesis during a given time interval can then be calculated as per Equation (1).

### *5.3.3. Light utilization*

Although the rectangular hyperbola (Equation (2)) can be used to obtain a good approximation of the photosynthetic rate per leaf, this rate will vary depending upon the location of the leaf in the canopy, which subsequently affects the value of $I_l$. Thus, an estimation of the total photosynthetic rate per plant must take into consideration the shading of leaves lower in the canopy. Canopy photosynthesis models are usually modifications of the leaf photosynthesis equation that give an estimate of crop or vegetative photosynthesis rates per square meter of ground cover (Thornley and Johnson, 1990; Acock et al., 1978). The simplest approach to modelling whole plant (as opposed to canopy) photosynthesis is to vary the value of incident light that reaches leaves in different parts of the plant, and to then use the resulting attenuated values of irradiance in Equation (2). Beer's Law (Thornley and Johnson, 1990, p. 243) is most commonly used to describe the degree of light attenuation by leaf cover:

$$I_l = I_o e^{-kL} \tag{7}$$

where:

$k$ = light extinction coefficient, $m^2_{leaf}\cdot m^{-2}_{ground}$, $0<k<1$
L = cumulative leaf area index at position in canopy, $m^2_{leaf}\cdot m^{-2}_{ground}$
$I_l$ = leaf irradiance at cumulative leaf area, L, $W\cdot m^{-2}_{leaf}$ of PAR

To estimate the irradiance received by plants in the ecosystem model described here, the vegetation profile is divided into three canopy levels (Figure 5.2). Tall plants such as trees are in Level A, shrubs and bushes are in Level B, and herbs, grasses and all young plants are in Level C. No plants can be in more than one canopy at a time (i.e., all of the foliage on a plant is assumed to be in the same canopy level), although some plants, such as trees, will be in different levels of

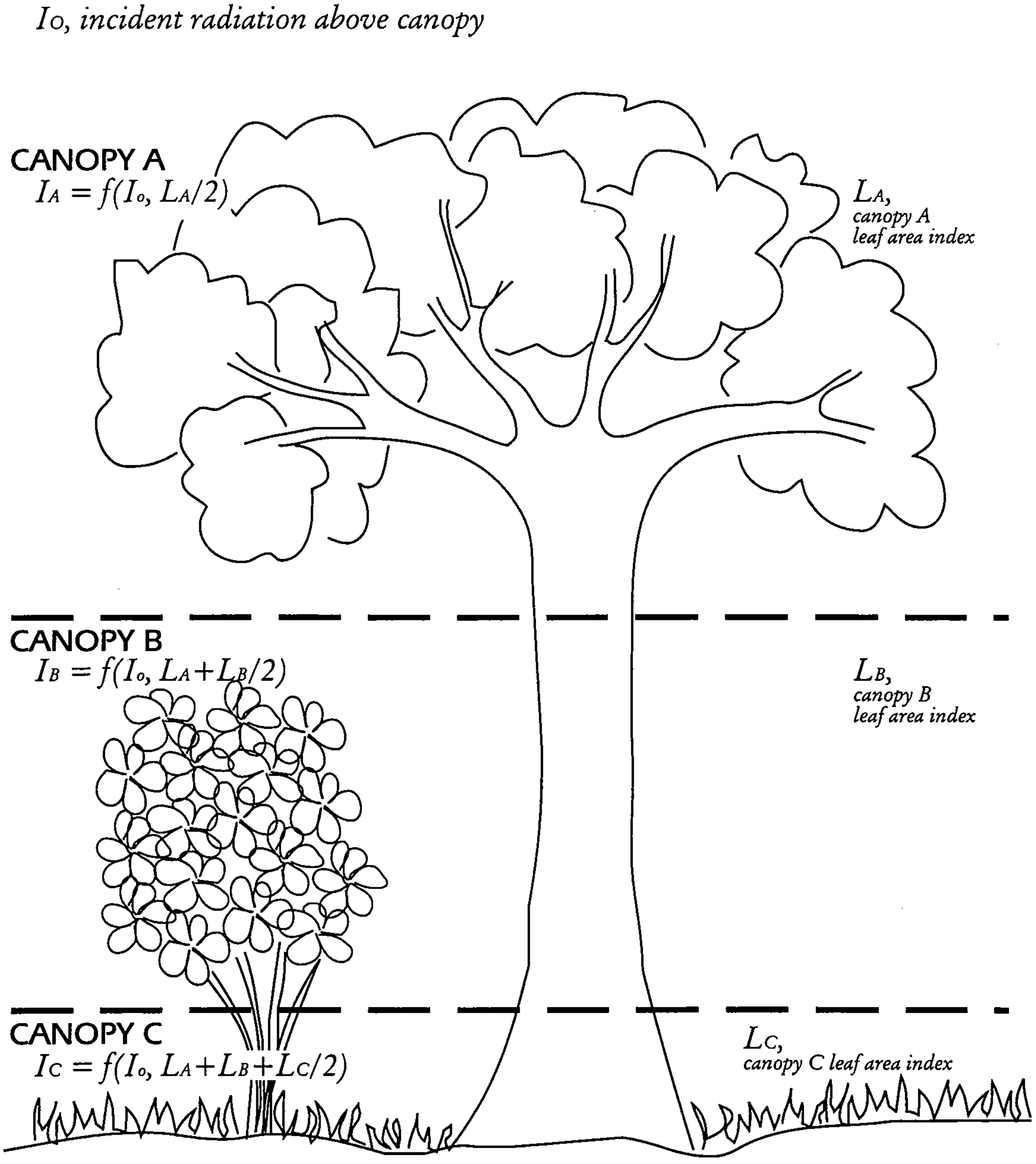


Figure 5.2: Canopy levels in the model ecosystem.

the canopy at different stages of growth. For each canopy level, in each grid cell, a total leaf area is calculated by summing the leaf areas of all the plant objects in that canopy (for each plant, the leaf area is calculated by dividing the total leaf mass by the leaf thickness; both are stored as plant attributes). A leaf area index ($L_A$, $L_B$ or $L_C$) is then computed for each canopy by dividing the total leaf area by the grid cell area. The irradiance on any plant at a given canopy level is then calculated as a function of the leaf area index in the canopy above, plus one half of the leaf area index of its canopy, using Beer's Law. Thus:

$$I_A = I_o e^{-k*L_A/2} \tag{8}$$

$$I_B = I_o e^{-k*\left(L_A + L_B/2\right)} \tag{9}$$

$$I_C = I_o e^{-k*\left(L_A + L_B + L_C/2\right)} \tag{10}$$

The values of $I_A$, $I_B$ and $I_C$ are used in Equation (2) when calculating the photosynthesis rates for plants in the different canopies[1]. All of the leaves on a plant are assumed to receive the same level of irradiance, as calculated according to Equations (8), (9) or (10).

### 5.3.4. *Respiration*

The basic equation that was used to model respiration is:

$$6CH_2O + 6O_2 \text{ ---> } 6H_2O + 6CO_2 + \text{energy} \tag{11}$$

Some of the carbohydrate produced by photosynthesis is used for respiration.

---

[1] There is one exception, intended to adjust for single plant objects that are not substantially affected by shading due by surrounding plants: If the total leaf area in the canopy above a plant, plus half of the leaf area in the plant's canopy is less than or equal to four times the plant's leaf area, then the amount of incident irradiance is calculated using the plant's leaf area index, NOT the canopy leaf area index. Thus, in such cases, the plant leaf area index (calculated as the plant's total leaf area divided by its "footprint", i.e. the ground surface area it shades) is substituted into Equation 8, 9, or 10, and the corresponding irradiance value is used in Equation (2) to compute P.

The net rate of mass gain due to photosynthesis is, therefore, modelled as being equal to the final gross photosynthetic rate ($P_{final}$) minus the respiration rate.

In real plants, the amount of photosynthates lost to respiration varies greatly with plant type as well as with the season and time of day. For the purposes of the model, however, a simple method, in which neither seasonal nor diurnal fluctuations are taken into account, is used to estimate the respiration rate of a plant. With this method, respiration is assumed to have two components, one proportional to the plant's photosynthesis rate, and the other a function of its dry mass:

$$R = (1-c_1)P_{final}A_{leaf} + c_2W_d \qquad (12)$$

where:

$R$ = respiration rate, $kg_{CO_2}\cdot s^{-1}$
$c_1$ = constant ($0 < c_1 < 1$), no units
$c_2$ = constant ($0 < c_2 < 1$), $kg_{CO_2}\cdot(kg\cdot s)^{-1}$
$W_d$ = dry mass of plant stem and root, kg

This approach is a modified version of McCree's equation (Hansen and Jensen, 1977; Thornley and Johnson, 1990), which was developed to describe the relationship between maintenance respiration requirements, plant dry mass, and conversion efficiency. This equation has been used widely to model the respiration of a wide variety of plants of different species. In the model described here, values of $c_1$ and $c_2$ are selected fairly arbitrarily for each plant species and are assumed to be constant throughout the life of a plant. Also, only the dry mass of stem and root parts is used to calculate the respiration cost of maintenance in order to keep this portion within reasonable limits for all plants. Note, also, that plants continue to respire to meet maintenance costs even when they are not photosynthesizing (e.g., during the night). If a plant does not have readily available carbohydrate for respiration (i.e., that created via photosynthesis), it will obtain some from organic storage.

### *5.3.5. Protein synthesis*

If the rate of photosynthesis exceeds that of respiration, there is a net accumulation of carbohydrate assimilate in the plant object. A portion of this is immediately converted to protein according to the following "made-up" reaction:

$$3N_2 + 6H_2O + 2C_6H_{12}O_6 \dashrightarrow 3C_4H_{12}O_4N_2 + 3O_2 \tag{13}$$

In this process, nitrogen from the inorganic storage and water from the leaves is combined with a fraction of the carbohydrate assimilate to produce protein and oxygen. The oxygen is then released into the atmosphere. The mass of carbohydrate to be converted to protein is calculated in a teleonomic manner, such that the plant maintains its optimal dry matter composition (i.e., carbohydrate:protein ratio).

### *5.3.6. Partitioning and translocation of assimilates*

After respiration and protein synthesis have occurred, the resulting mass of carbohydrate and protein constitutes the plant's net assimilate production (i.e., mass growth) over a given unit of time. These assimilates are subsequently distributed amongst the various parts of the plant in accordance with a teleonomic partitioning scheme in which each growing part "attempts" to attain a mass equal to a pre-specified fraction of the plant's total mass by harnessing an appropriate share of the assimilate to be distributed.

For each modelled plant species, the year is divided into two periods: dormancy and growth. Mature plants (i.e., those not in seedling or senescence life stages, see Section 5.3.7 for a description of the life cycle stages modelled) also undergo a reproductive phase during, and usually at the end of, the growth period. For each period or phase, target fractions are assigned to each plant part (leaf, stem, root, seed, etc.). Leaves on a deciduous plant, for example, might be assigned a target fraction of 0.0 during the dormant period, and a fraction of, perhaps, 0.2 during the growing season. These fractions may be modified slightly during the reproductive phase of the growth period to accommodate new seed mass. In addition, seedlings and senescing plants may be assigned slightly different target fractions than mature plants: a seedling, for example, would probably have a much higher leaf:stem ratio than a mature plant. In the model, all of these target fractions, for the different life stages and phases of growth, are assigned as species-specific attribute values.

The partitioning of assimilates is carried out by comparing the actual mass fraction of each plant part (e.g., $m_{leaf}/m_{plant}$) with the target fraction. If the actual mass fraction is lower than the target fraction, then that part is a growing "sink". The strengths of all of the "sink" parts on a plant are calculated, and then weighted relative to one another, so that the sum of all "sink" strengths is equal to one. Thus, if only one part were currently a "sink", then its strength would be equal to one. The assimilates (newly formed carbohydrate and protein) are then distributed to the carbohydrate and protein components of the various plant parts according to their relative "sink" strengths. (Within plant translocation is assumed to be instantaneous.) The net result of this partitioning scheme is regular overall plant growth that is appropriately allocated to the various plant parts over the year. This is illustrated in Figure 5.3, in which the mass of each part of a single mature plant is plotted over a one-year period. The target fractions assigned for this species are shown in Table 5.2.

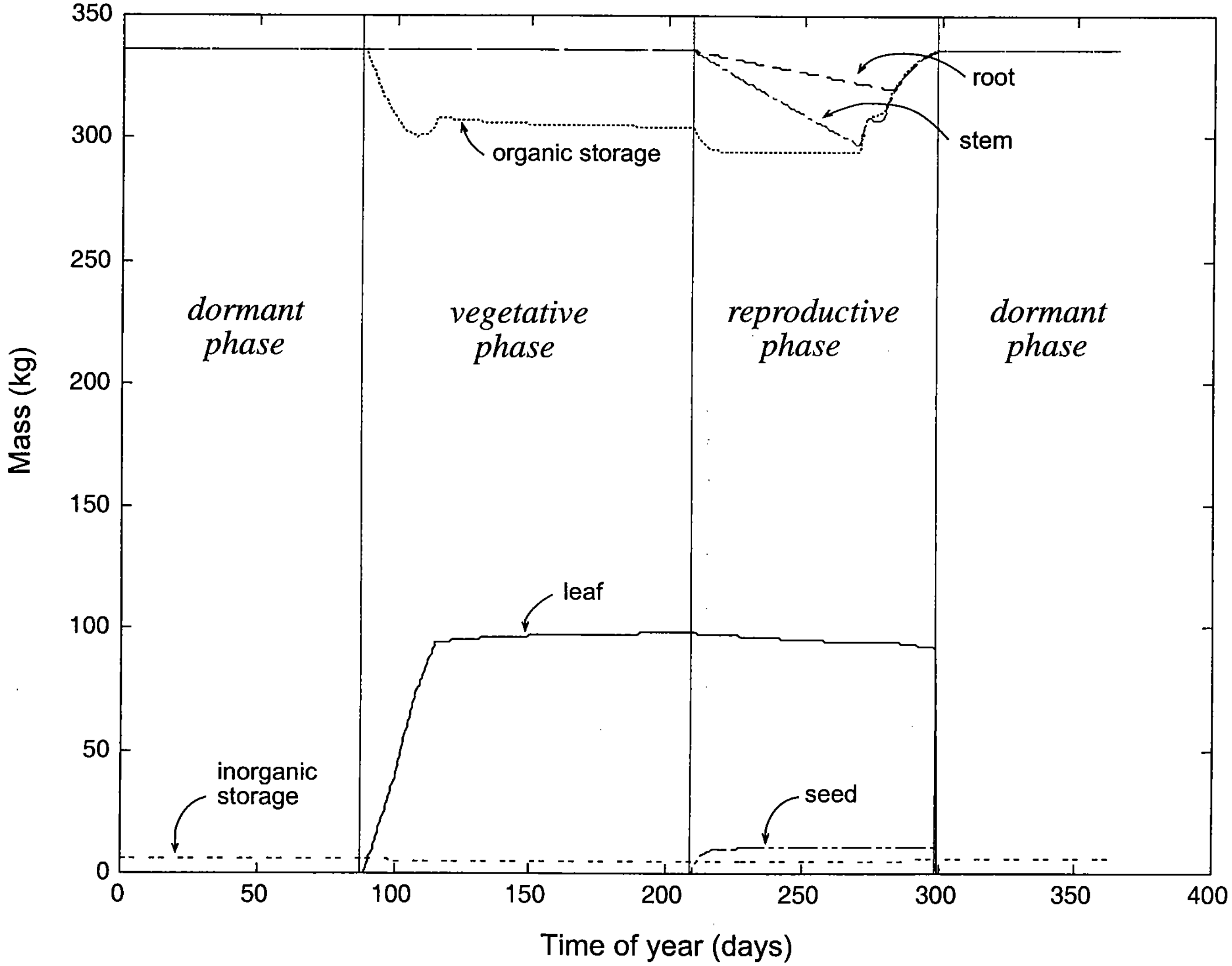


Figure 5.3: Masses of the parts of a typical mature deciduous tree object during one simulated year.

| Plant part | Dormant phase | Vegetative Phase | Reproductive Phase |
|---|---|---|---|
| leaf | 0.0000 | 0.0900 | 0.0891 |
| stem | 0.3297 | 0.3000 | 0.2970 |
| root | 0.3296 | 0.3000 | 0.2970 |
| organic storage | 0.3350 | 0.3050 | 0.3020 |
| inorganic storage | 0.0057 | 0.0050 | 0.0050 |
| seed | 0.0000 | 0.0000 | 0.0099 |

Table 5.2: Target fractions specified for each part of the mature, deciduous tree illustrated in Figure 5.3.

For each time step during which a plant photosynthesizes, if there is a net production of photosynthates, these must be assimilated into the plant. If there are no "sinks" (i.e., all of the plant parts are at their target fraction of the plant's total mass), then the new assimilate is added to the plant's organic storage. This serves to drive new growth, since the increase in organic storage mass will cause all of the other plant parts to subsequently become "sinks" into which further assimilates will be accumulated. The model plants will continue to grow throughout their lives until they reach a balance between leaf photosynthesis and maintenance costs.

The organic storage plays an important role in plant development, both at the beginning of each annual growth period and during the reproductive phase. At these times, carbohydrate and protein from this reserve are used to augment that which is available due to photosynthesis, in order to support the rapid growth of new leaves or the production of fruit. The organic storage mass is replenished late in the growing season when photosynthesis rates are high and the growth rate is relatively moderate (Figure 5.3).

### 5.3.7. *Life cycle stages*

A number of stages in the life cycle of a plant object are accommodated in the model (Figure 5.4). They are: seed, seedling, mature plant, and senescing plant. Except for the seed stage, the durations of each of these are attributes of a given species, as are the times of year that these may occur. Seed germination is a function of temperature and day length; after this occurs, a plant is assumed to graduate from stage to stage according to the passage of time. If a seed has not germinated after a maximum length of time, it is assumed to become unviable and is assimilated into the soil. Thus, a perennial plant may spend several years as a seedling, after which it will progress to the mature plant stage, and cycle through vegetative, reproductive and dormant phases, until senescence. Although many real plants only produce seeds in intermittent years or according to climatic conditions, in the model, plant objects reproduce every year. Senescent plants may stay alive for several years but, like a seedling, do not have a reproductive phase, and thus do not produce seeds. Annual plants must pass through their stages much faster than perennial ones, having to complete an entire life cycle in a single growth season.

### 5.3.8. *Litter production*

A mature plant has maximised its leaf area index to a point such that any additional leaf and stem growth will negatively impact its overall production by increasing respiration costs (Charles-Edwards et al., 1986). Thus, real trees and bushes abscise any limbs whose net production of photosynthates is negative over the growing period. These are usually the shaded branches that are lower in the canopy. Plants that have reached their maximum size (and, therefore, their maximum standing mass) may, therefore, continue to produce new leaves, but will drop older leaves at the same rate as new ones are produced (given constant environmental conditions). This is an important strategy on the part of plant in terms of maintaining its health and productivity, and results in the production of litter which becomes part of the soil's decomposing plant matter. In the model used here, plant mass loss due to litter production is described with

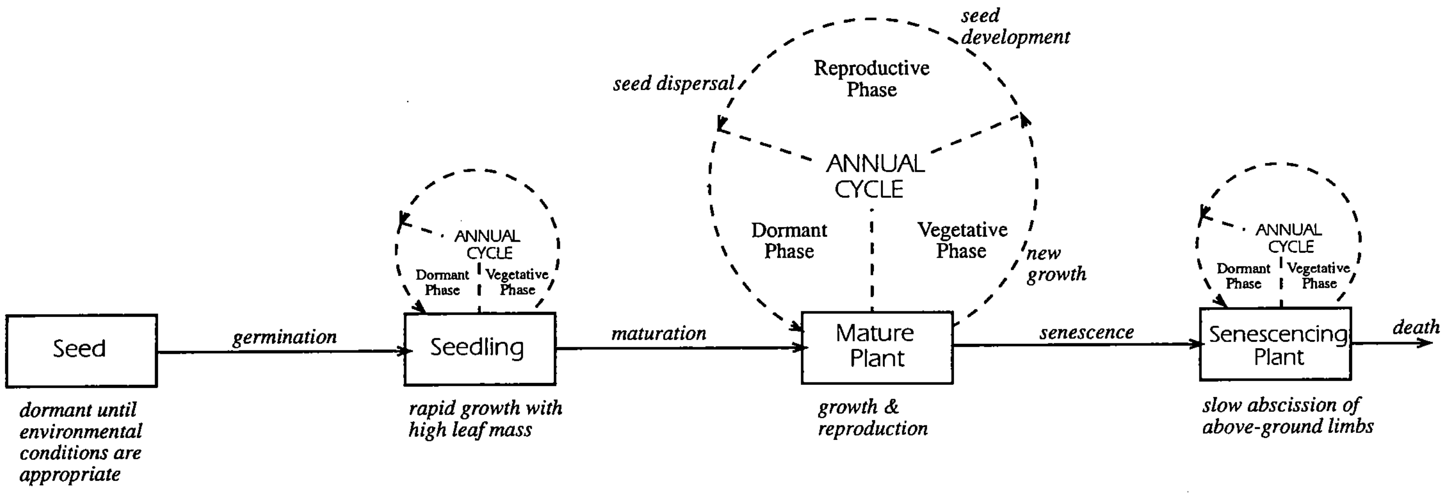


Figure 5.4: Stages in the life cycle of a plant.

first-order kinetics, i.e., at each time step all mature plants loose a certain fraction of their leaf, stem, and root mass to litter (the rate constant is a species-specific attribute value). In addition, after a plant reaches its maximum allowable mass (also a species attribute value), its loss rate is increased so as to equal its rate of production of new material.

Aside from the regular abscission of limbs, other forms of litter production are also incorporated in the model. For example, deciduous plants abscise all of their leafy material at the end of the growing season, and any plant may do so given a sufficient degree of environmental stress (e.g., freezing or dehydration). Evergreen plants are assumed to lose leaves or needles at a constant rate over the year due to aging. In addition, perennial grasses and herbs have a very high turnover rate of leaf and stem material, and may lose the majority of their above-ground vegetative mass at the end of the growing season.

### *5.3.9. Plant-water relationships*

In the model, plant objects intake water through their roots as required to maintain their total water content at the optimal level. They are restricted in their water usage at any given time in that, (1) the unsaturated water content of the terrain in the plant's grid cell must be greater than or equal to the amount of water required by the plant, and, (2) there is a limit to the amount of water a plant is capable of absorbing in a given unit of time, specified by the species-specific water absorption rate attribute ($\mathrm{kg_{H_2O} \cdot kg^{-1}{}_{root} \cdot s^{-1}}$). Water is not recovered from plant parts before they are dropped, nor is water flow to senescing plant parts ceased.

Whereas water enters a plant via the roots, it regularly exits via evaporation from the surface of the leaves. In the model, the amount of water evaporated from the surfaces of plant leaves is estimated using a simple linear function:

$$Z = Z_m \left( \frac{P_{final}}{P_m} \right) \tag{14}$$

where:
$Z$ = net evaporation rate, $\mathrm{kg_{H_2O} \cdot s^{-1} \cdot m_{leaf}^{-2}}$
$Z_m$ = maximum evaporation rate, $\mathrm{kg_{H_2O} \cdot s^{-1} \cdot m_{leaf}^{-2}}$

The maximum evaporation rate, $Z_m$, is input as a species-specific attribute value. More sophisticated single plant transpiration models (e.g., Friend, 1995) did not prove to be applicable to the wide range of plant types, nor the level of resolution, represented by our model. This equation, therefore, is used to provide a rough estimation of the effect of environmental conditions on water lost from vegetation to the atmosphere. Each modelled plant species is assigned a value indicating its relative tolerance to water stress; a plant that loses an excessive amount of water from its leaves and is unable to replenish this will first wilt and later die, in accordance with the stress to which it is subjected.

### *5.3.10. Nitrogen use*

As described above, nitrogen is utilized by the plants to synthesize protein from carbohydrate assimilate. To meet the need for this as it arises, a small supply of inorganic nitrogen is assumed to be stored in each plant's permanent tissues (this is the plant's "inorganic storage" part (Figure 5.1)). If this supply is low, a plant will attempt to replenish it with nitrogen absorbed from the inorganic nitrogen pool in the soil at the location in which it is growing. If there is inorganic nitrogen available, the plant can absorb as much as it requires (so long as there is a sufficient amount in the soil), at a rate equal to the root's intrinsic absorption capacity, expressed as $\mathrm{kg_{N_2} \cdot kg_{root}^{-1} \cdot s^{-1}}$. The absorption rate is not affected by the concentration of available nitrogen in the soil.

Since nitrogen tends to be a limiting resource, most natural plants will recover it from senescing plant parts before abscission. Thus, a real plant may translocate up to 75% of the nitrogen in leafy material before it is dropped. In models of N recovery from abscising plant shoots, a value of 50% is often used as an average (Friend et al., 1997). In this model, during the final part of the growing season, deciduous plants and herbaceous perennials recover between 30% and 70% of the nitrogen from their leaves by breaking down proteins at a constant rate, the value of which is a species-specific attribute ($kg_{N_2} \cdot s^{-1}$) (using a process that is the reverse of protein synthesis, see above). The captured nitrogen is placed in inorganic storage and is used to produce new protein mass in the next growing season. Nitrogen is not recovered from plant parts that are dropped due to routine, daily litter production.

### *5.3.11. Seed dispersal*

During its reproductive phase, a mature plant grows seeds (this is its "seed mass", Figure 5.1), which are then dispersed at the end of this phase to create new plants. Vegetative reproduction is not currently included in the model. Herbaceous plant objects such as grass do, however, increase in area as their mass increases until they fill the entire grid cell. Thus, their spread (and, therefore, their maximum mass) is bounded, somewhat unrealistically, by the "edges" of the cells in which they reside. This, however, is not a major drawback since the seeds can be easily dispersed to neighbouring cells.

The number of seeds created by a modelled plant is determined based on the magnitude of its seed mass, and the value of the mass of a single seed for a plant of its species (this value is stored as an attribute for each plant object). To avoid overloading a simulation with unmanageable numbers of individual plants (both in terms of memory and computational requirements), a large portion of the seeds are never given the opportunity to germinate, and are simply added to the soil in the grid cell of the parent plant. These seeds are then available for consumption by gramnivorous species, and will ultimately decompose should they not be consumed. For individually modelled plant species (i.e., those for which a plant object is an individual plant), a small proportion of seeds become new plant objects of development stage "seed". As described above, these have the opportunity to germinate should conditions be favourable. The new plant objects are distributed over a circular area with the parent plant at the centre. The direction in which each seed falls is selected randomly, and the distance at which it falls (in grid cells) is calculated according to a normal distribution having a mean of zero, and a standard deviation equal to 0.67 of the circle's "maximum" radius. The maximum radii of the distribution circles are 49, 10 and 2 grid cells for plants in canopies A, B and C respectively.

Viable seeds for lumped plant species are distributed slightly differently. To help cap the total number of individuals represented by the model, only one instance of each lumped species is permitted per grid cell (it may, however, cover the entire surface area of the cell). Each lumped plant object has an associated "ungerminated seed mass", which is a mass of seeds that may germinate if conditions are appropriate. If a viable seed of a lumped species is distributed onto a grid cell in which there already exists an instance of that species, the seed's mass is simply added to that object's "ungerminated seed mass". If, however, seeds of a lumped species fall into a grid cell in which that species is not yet growing, a new instance of that species is created which, in this case, exists only and entirely of the ungerminated seed mass (i.e., the plant object is in the "seed" stage). For annual plants that die at the end of the growing season, the ungerminated seeds remain and the plant object reverts to development stage "seed" in the following season. If a plant object is perennial and lumped, its "ungerminated seed mass" is simply incorporated into the plant's organic storage mass at the beginning of the growing season, or whenever environmental conditions are favourable. If environmental conditions are unfavourable, a perennial plant's ungerminated seed mass is added in small fractions over time to the soil.

### *5.3.12. Response to environmental stress*

Plants may be subjected to a number of different environmental stresses during their lifetimes including drought, lack of light, predation, or extreme temperatures. Each of these is addressed in the model with a mechanistic approach, in which a plant's response is the natural result of its attempt to continue normal growth. Thus, extreme temperatures or low light conditions will be reflected in severely decreased, or completely halted, photosynthetic activity. If these conditions persist, the plant will respire away all of its organic storage in order to maintain itself and will eventually die. A plant that has a very low water content due to drought conditions will first wilt (and drop its leaves) and then, if the condition persists, it will die as a result of the subsequent lack of photosynthetic activity. The cost of predation on a plant will vary according to its severity. Herbivorous animals in the ecosystem will consume the stem and leaves of most plant species. A plant that loses only small fractions of these parts will recover fairly easily. A plant that loses most of its leafy material to predation, however, will experience stunted growth due to lower photosynthate production, and will be constantly allocating any new assimilates to the regrowth of affected areas.

### *5.3.13. Nitrogen fixation*

Although, in physical ecosystems, the fixation of atmospheric nitrogen is carried out by microorganisms, often in symbiosis with plants, in this model it is assumed to be a plant function. Thus, selected plant species are assumed to have the "ability" to fix nitrogen at a rate that is a function of the plant's total root mass, and its current photosynthesis rate:

$$N_{fixrate} = r_n m_{root}\left(\frac{P_{final}}{P_m}\right) \tag{15}$$

where:

$\mathrm{N_{fixrate}}$ = rate of nitrogen transfer from atmosphere to soil, $\mathrm{kg_{N_2}\cdot s^{-1}}$
$r_n$ = rate constant, $\mathrm{kg_{N_2}\cdot(kg_{root}s)^{-1}}$
$\mathrm{m_{root}}$ = plant root mass, kg

### *5.3.14. Plant-plant interactions*

Plants in the same grid cell compete for light, water, and nutrients. As previously mentioned, the plant canopy is vertically differentiated into three layers. Thus, plants such as herbs and young trees growing lower in the canopy are shaded by taller plants above. All plants in a grid cell compete equally for access to water and nitrogen from the soil, since root depth is not taken into account, and there is no vertical stratification in the soil. There is some secondary competition between plants in adjacent (and further) grid cells, since there is subsurface water flow between these. For instance, water may flow from neighbouring cells into a cell containing a large tree that is regularly draining water from the soil.

## 5.4. Verification of the plant model

The performance of the plant model was tested first in isolation, in which single plant objects of each species were simulated under conditions of unlimited water, nutrient, and light availability (except that the leaves were subject to shading by other leaves on the plant). These tests were used to assess and select values for all of the various plant object attributes, and to ensure that the model performed as expected. The deciduous tree shown in Figure 5.4 was generated with such a test. Next, the plant model was tested in the context of the overall ecosystem model which includes representations of a terrain and atmosphere as well as biological components, all of which are subjected to climatic forcing functions. At first, for each species, only a few instances were added to the ecosystem, and the attributes of each were recorded for several simulated years, and inspected to verify the model's performance. The plant model was used in larger ecosystem simulations only once a reasonable degree of confidence in its performance was achieved.

## 5.5. Performance of the plant model in ecosystem simulations

Overall, the plant model provides for considerable flexibility, and is currently being used in simulation to represent a number of plant species, in a variety of different ecosystem configurations, including grassland and woodland type biomes. All of the simulations reported here were executed with the same series of temperate climate forcing functions, and the same initial encompassment configurations (e.g., terrain size and topography, soil composition, atmospheric conditions, etc.). Only the configuration of biological components (included species and initial population sizes) have been varied, in order to illustrate the effectiveness of the plant model in predicting vegetation dynamics[1]. A discussion of the global effects of vegetation on abiotic factors in the ecosystem is presented in Parrott and Kok (2000b). The results given below are intended to illustrate some aspects of multi-species population dynamics that emerge from the combined activities of many plant objects.

### *5.5.1. Biomass accumulation*

The photosynthesis rate of a plant object, and therefore the mass that it can attain, is partially determined by the availability of light. The latter is a function of the incident radiation, as well as shading by neighbouring plants (i.e., the canopy's leaf area index). Thus, for a given ecosystem configuration and set of forcing functions there should be a maximum possible total amount of living plant biomass that can accumulate per unit area (this is the "specific biomass" in $kg_{\text{dry biomass}} \cdot m^{-2}$). In Figure 5.5, the specific biomass for two different simulations is shown for a period of 70 years[2]. In the first of these, the model was configured to represent a "grassland" ecosystem, consisting of six species of perennial and annual grasses and herbs. This system attained a fairly stable living biomass in 20-25 years, in which the dry biomass per unit area ranged seasonally between 1.75-4.5 $kg \cdot m^{-2}$, which falls within the normal high and low specific biomass values reported for terrestrial temperate grassland ecosystems (i.e., 0.2-5 $kg \cdot m^{-2}$; Whitakker, 1975). In the second simulation, the model was configured with similar initial populations of grasses and herbs, together with several tree species

---

[1] It should be noted that for each simulation, the ecosystem was initialised with plants in both the seed and mature plant life cycle stages.

[2] Thesis note: These are the Baseline2 and Baseline3 simulations.

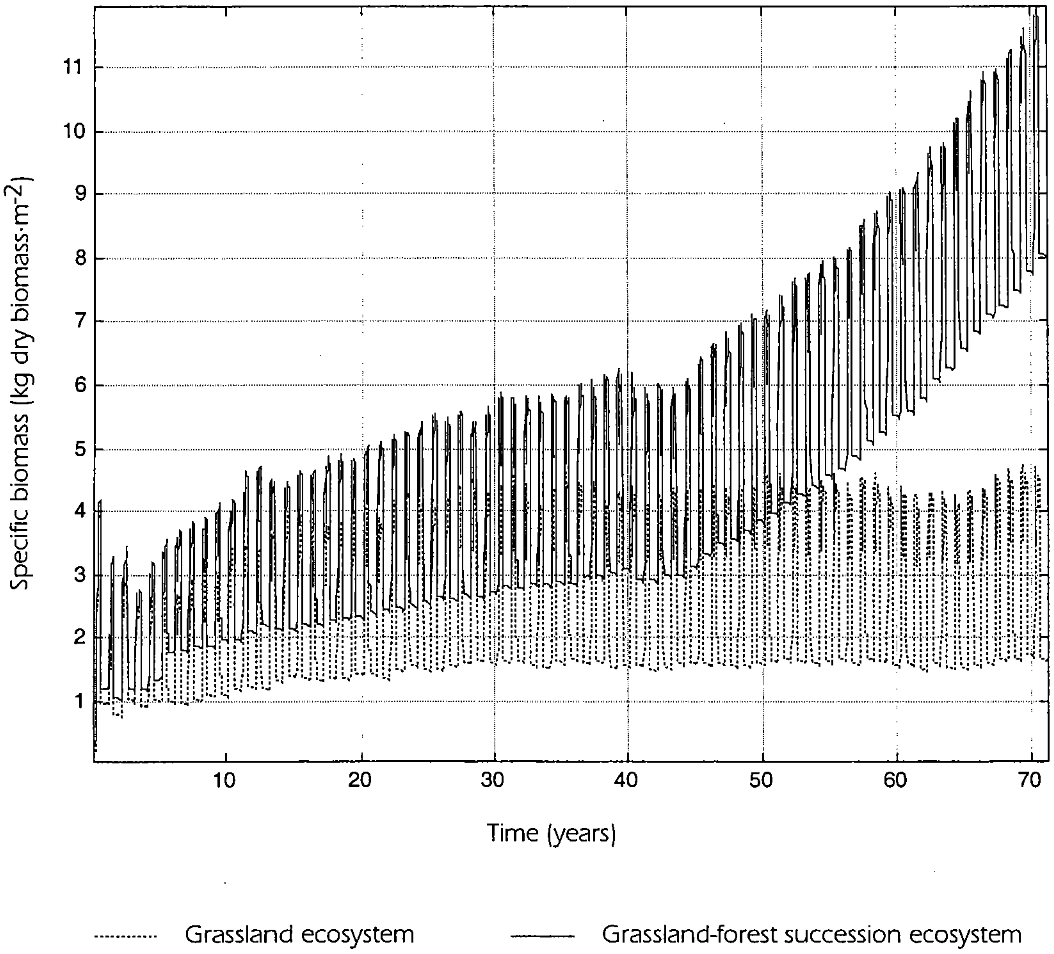


Figure 5.5: Specific biomass versus time for two ecosystem configurations.

(both deciduous and evergreen). The trees were initially few in number, and quite sparsely distributed, and the sequence of events that occurs is illustrative of a process of succession from fairly open grassland to forest. The specific biomass shows a steady, and later a sharp, increase, as the trees begin to overtake the herbs and grasses, thereby changing the landscape. This is still ongoing at the end of the simulation period, at which point the end-of-season specific biomass had reached about 12 kg·m$^{-2}$. Presumably, this would plateau after several more decades. Whitakker (1975) has indicated that a temperate climate forest would normally attain a specific biomass value in the range of 6 and 60 kg·m$^{2}$ at full maturity.

### *5.5.2. Plant population dynamics*

The population histories of four of the plant species from the grassland-forest succession ecosystem described above are shown in Figure 5.6. Most plant species, both those shown in Figure 5.6 as well as others, exhibit an annual variation in their population sizes. For lumped species, the total population size is artificially limited to the number of grid cells on the terrain, so their numbers tend to fluctuate somewhere below this maximum. Species such as trees that are not capped in their total population sizes, generally show an upward or downward trend in total numbers during short simulations, such as is shown in Figure 5.6(d).

Ecosystems configured with both herbs and trees often undergo a transition period, as was the case for the grassland-forest model (Figures 5.5 and 5.6), in which the tree species slowly overtake the herbs in both number and specific biomass. In many simulations, the transition is quite rapid: The sizes of several different herb and tree populations in a 25-year simulation are plotted in Figure 5.7. The encompassment and climate used for this simulation was identical to that of the others, the difference being the addition of a particularly productive tree species (i.e., in terms of seed production and seed viability), the population of which reaches close to 50,000 instances in 25 years. Note that the grass and herb populations begin to decrease in size as this dominant tree takes

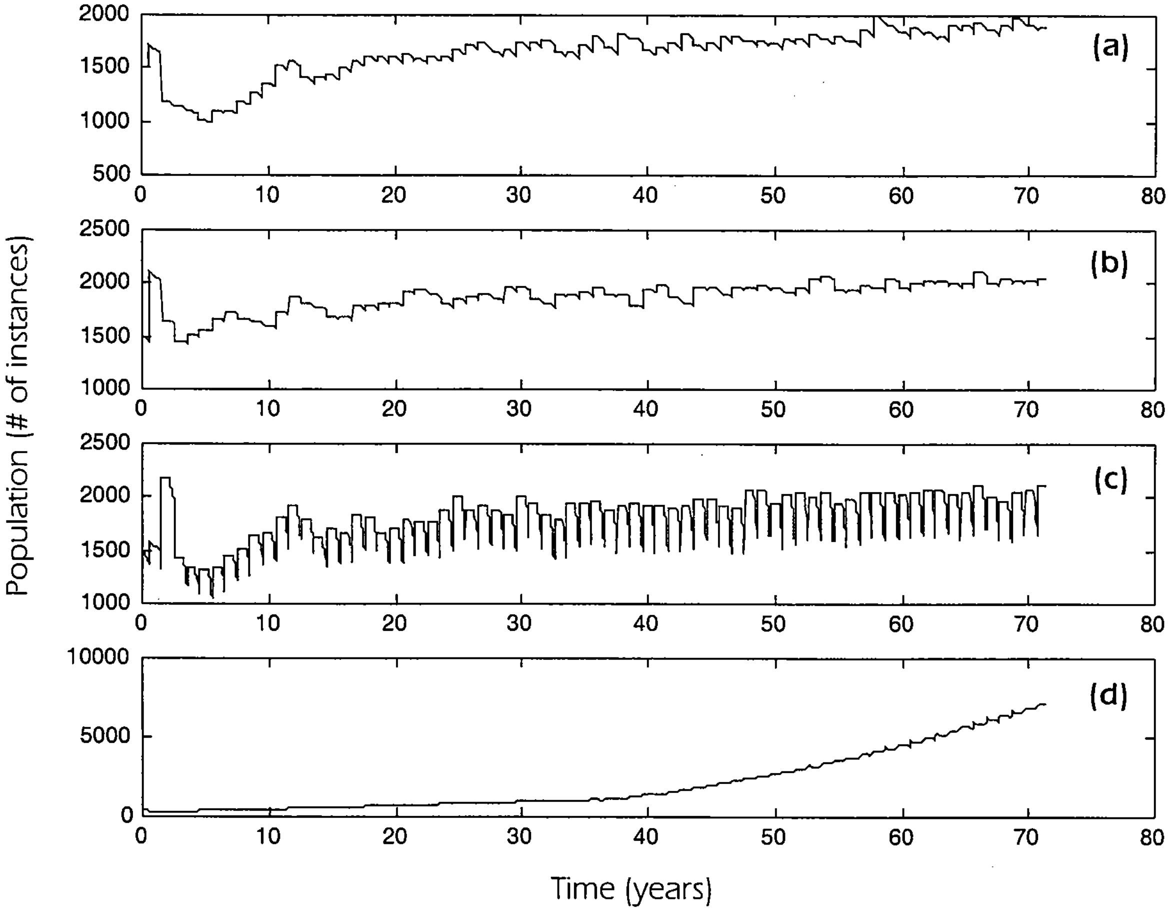


Figure 5.6: Population dynamics of four plant species in a model ecosystem (a) annual grass (#5320); (b) annual grass (#5156); (c) perennial grass (#5166); (d) deciduous tree (#5076).

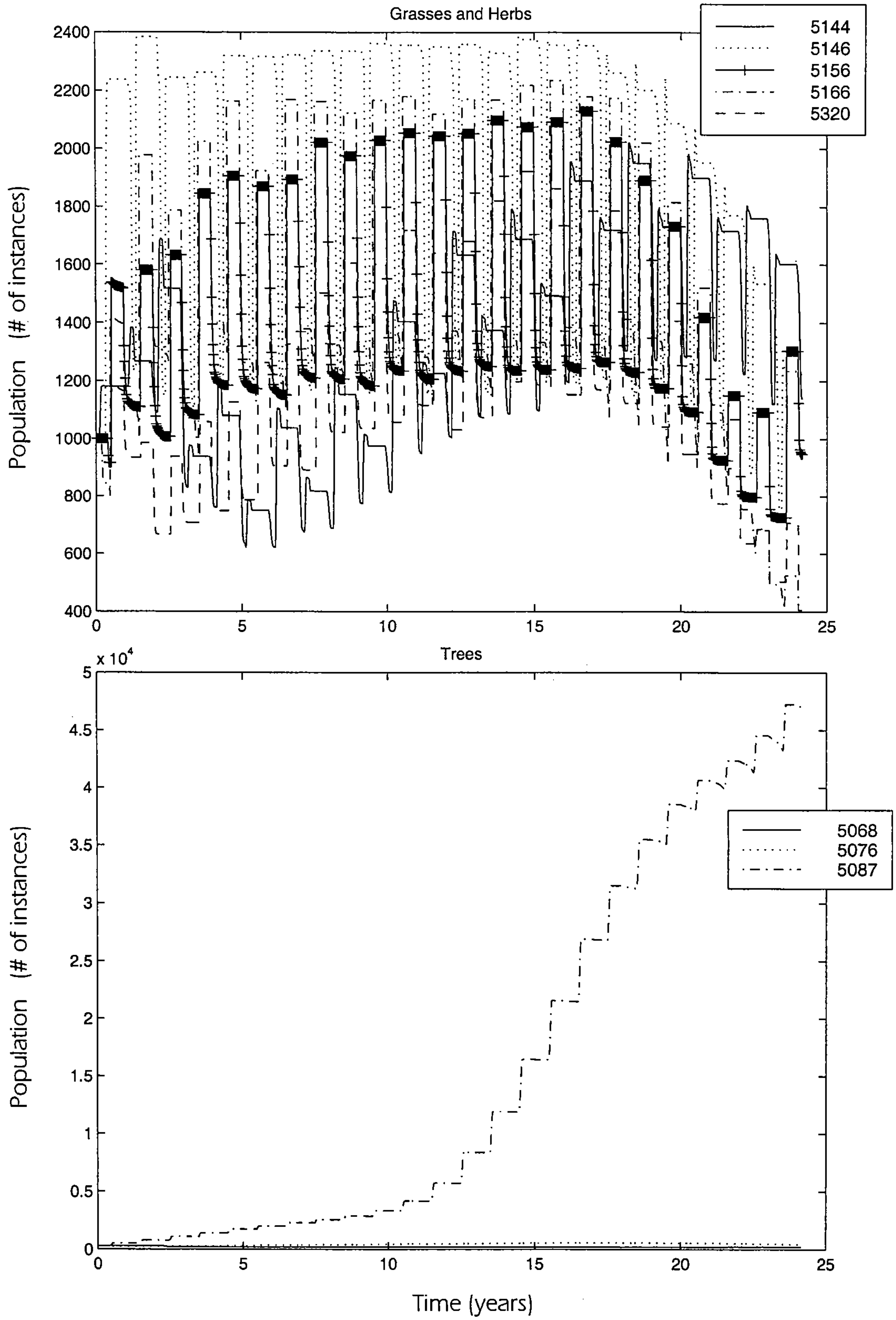


Figure 5.7: Populations of selected herb and tree species in an ecosystem undergoing succession from grassland to forest (numbers in legends refer to species #).

over near the end of the simulation.

### 5.5.3. *Response to grazing by herbivores*

All of the simulations described so far were executed for strictly plant-based ecosystem configurations. A number of experimental runs based on ecosystems that include herbivorous animals in addition to plants have also been executed. In these, the impact of grazing on the plant populations is substantial. Although most grazers do not consume the whole plant, many plants are extirpated due to substantial losses of above-ground leaf and stem mass and their corresponding inability to support respiration and re-grow. The effect of predation in a sample grassland ecosystem configured with five herbaceous species and two small herbivorous mammalian species is illustrated in Figure 5.8. In this experiment, one grass species became extinct (after 10 years), as did the larger of the two herbivore species (after about 8 years; this species had a larger body mass and therefore greater food intake requirements; it also had a lower reproduction rate). The smaller herbivores underwent substantial population crashes during the winter season, due to starvation. It appears that during the final few years of simulation, the herbivore population has reached the maximum size that can be supported by the plant community over the long-term.

### 5.5.4. *Spatial variation and patterns of distribution*

In real ecosystems, spatial variation and patterns of distribution of plants over a terrain occur as the result of a number of factors, such as: differences in the microclimates of different locales; between-plant competition for resources; and predation by other species. The degree to which plants of a particular species are affected by these factors depends to a great extent upon their physiology and other characteristics. In the ecosystem model, spatial variation and patterns of distribution of plants can occur due to any of these factors, although in the simulations reported here, the encompassment was not sufficiently differentiated, nor large enough, for there to be differences in microclimates (the terrain was only 500x500m). Thus, while the plants will respond differently to

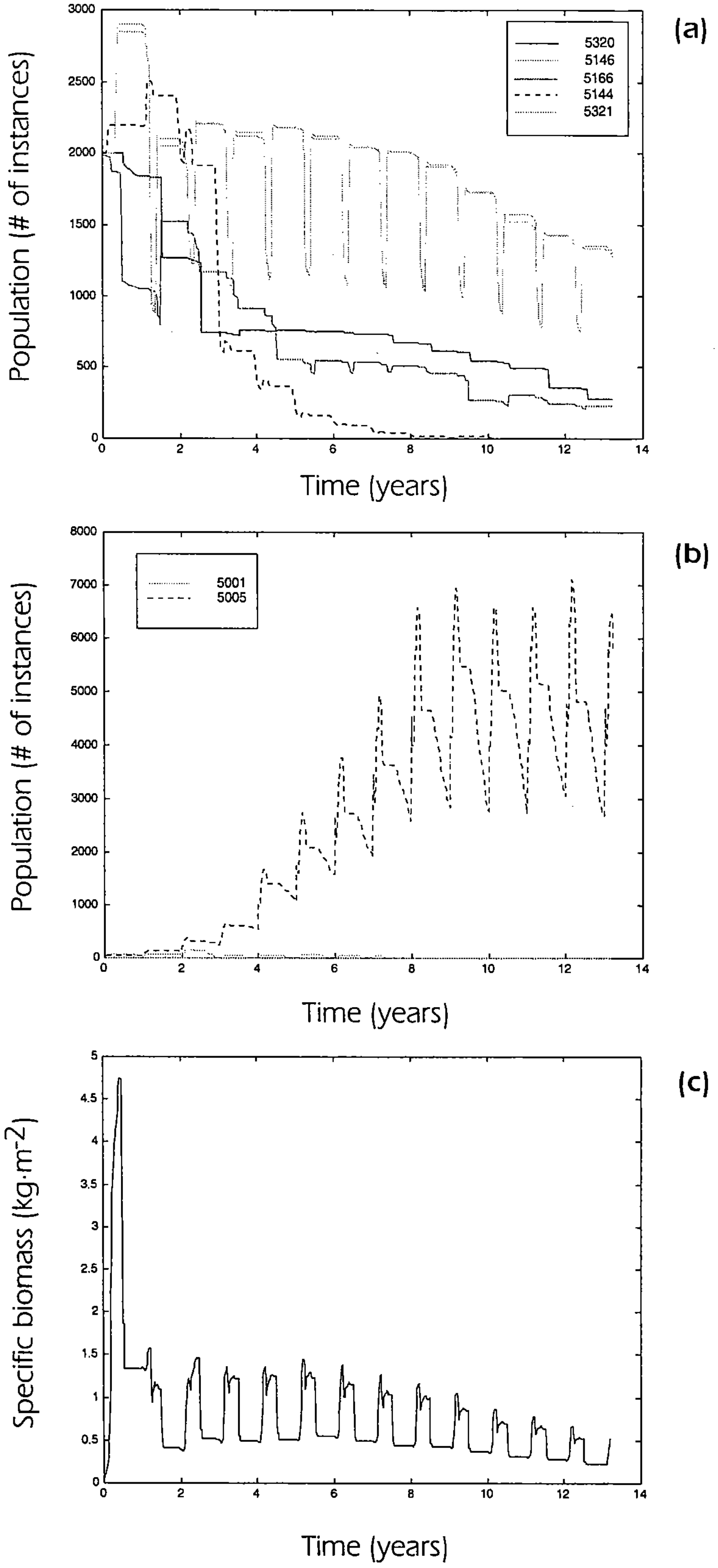


Figure 5.8: Effects of grazing by small mammalian herbivores on plants in a grassland ecosystem configured with 5 herbaceous species and 2 herbivorous species. (a) Populations of herbs and grasses; (b) Populations of herbivores; (c) Total specific biomass of the system.

different climatic environments, due to the small terrain size, this aspect was not studied in the context of spatial variation. The plant species do, however, exhibit patterns of distribution due to plant characteristics such as seed dispersal ranges, as well as inter-species competition, and grazing by herbivores. Some sample patterns are shown in Figure 5.9, Figure 5.10, & Figure 5.11.

For these figures, three plant species (an annual grass, a perennial herb, and a perennial grass) were selected for more detailed study. The distribution of these species over the terrain was observed for two different simulations. For the first of these, the ecosystem was configured as a grassland with only plant species (Figure 5.5), and for the second, the ecosystem was configured with similar initial populations of plant species, plus three species of small mammalian herbivores (Figure 5.8). The relative abundance (in terms of mass) of each species in each grid cell at the end of the growth period (late summer) was then plotted for 12 consecutive years, starting at Year 0 of the simulation. In each case, instances of every species were initially distributed randomly and uniformly on the terrain. The situation for the annual grass is illustrated in Figure 5.9, for the perennial herb in Figure 5.10, and for the perennial grass in Figure 5.11. In each of these figures, colouring of the squares indicates the presence of the species, and darker squares depict higher masses. With reference to the figures, the following observations can be made: First, in most of the cases (except for Figure 5.11(a)), the manner in which each plant species is distributed about the terrain changes from an initial, fairly homogeneous random distribution, to a more heterogeneous one in which clumping patterns can be discerned. Second, when there are clumps, they tend to persist from one year to the next. Third, there is a marked difference in the patterns that emerge from the two different ecosystem configurations. The effects of herbivory are clearly evident in Figures 5.9(b), 5.10(b), and 5.11(b), in which the distribution of each plant species becomes considerably more heterogeneous than it was without herbivory. Fourth, in both ecosystem configurations, there is a qualitative difference in the kinds of patterns that emerge for the different species. For example, the perennial herb (Figure 5.10) tends to form much more distinct clumps than the perennial grass

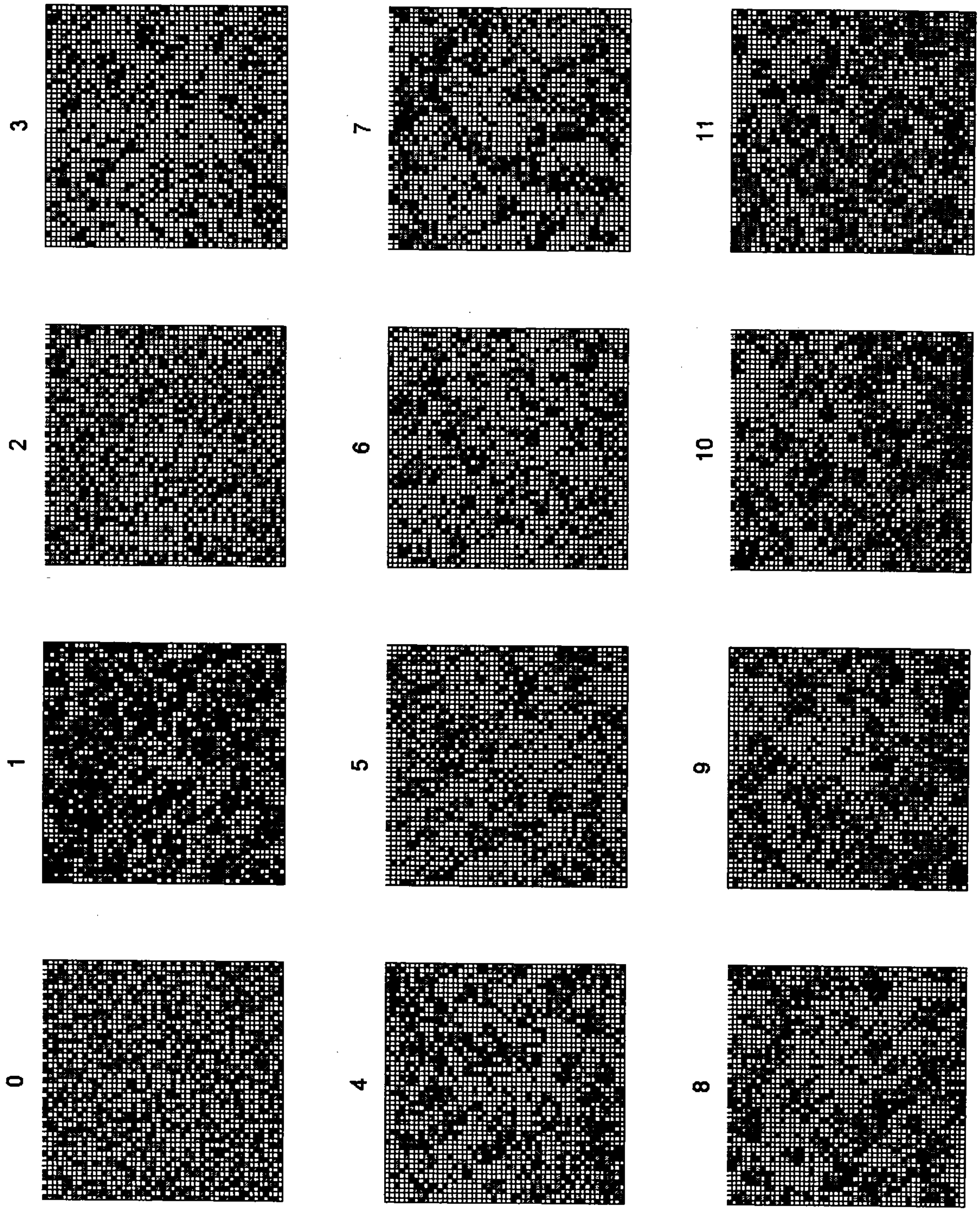


Figure #5.9(a): Relative abundance of an annual grass species (#5320) in a grassland ecosystem, no herbivory.

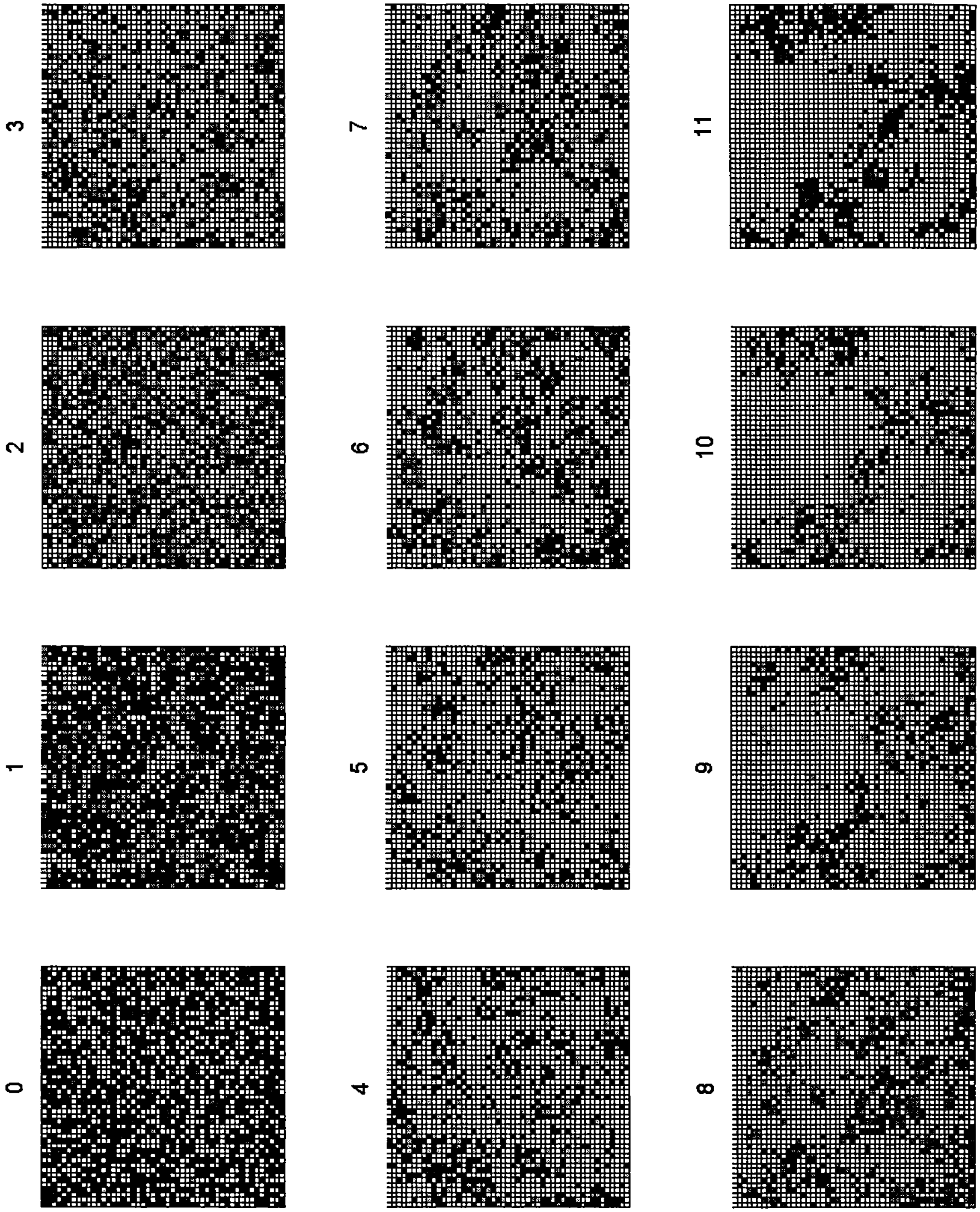


Figure #5.9(b) Relative abundance of an annual grass species (#5320) in a grassland ecosystem with small mammalian herbivores.

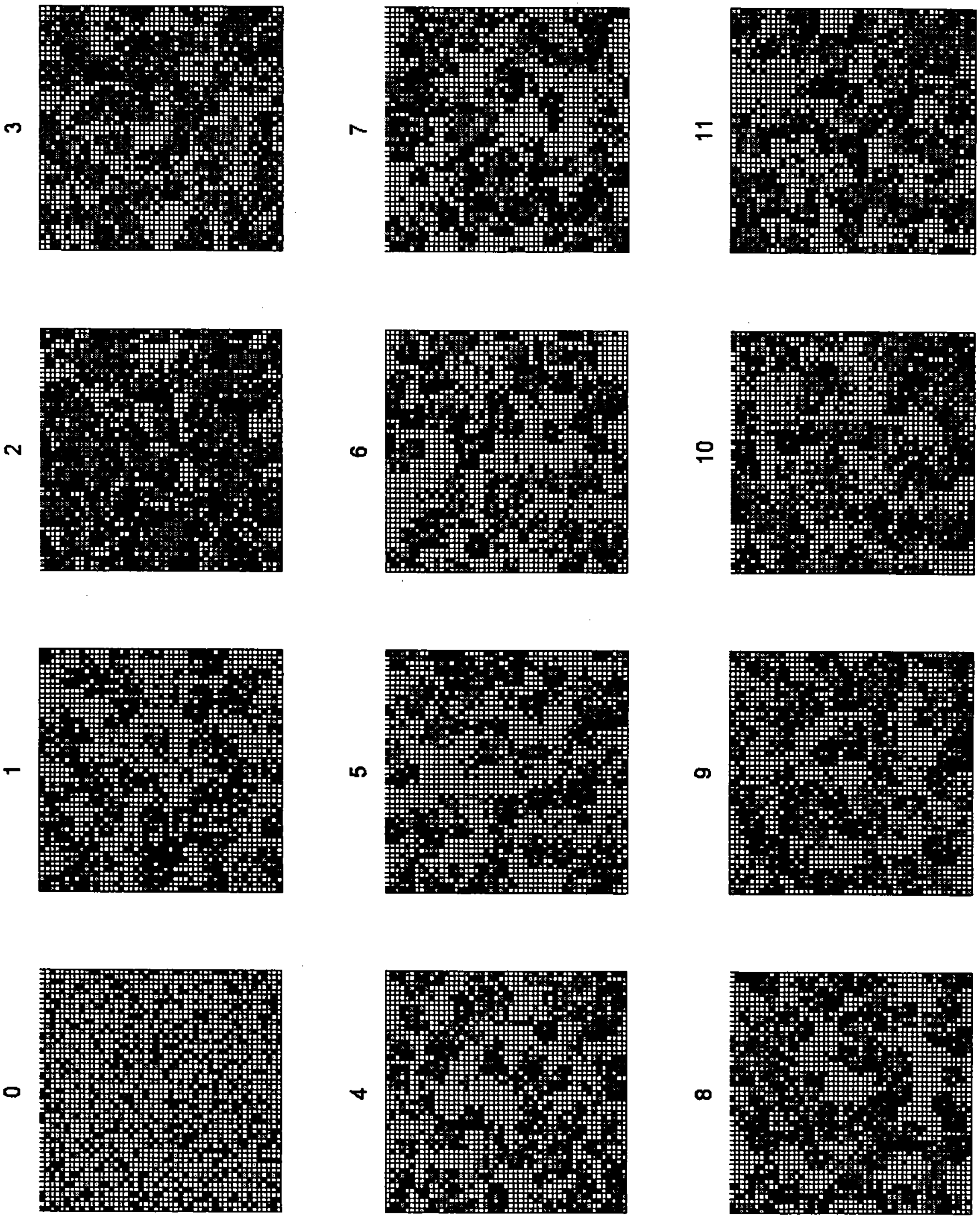


Figure #5.10(a): Relative abundance of a perennial herb species (#5166) in a grassland ecosystem, no herbivory.

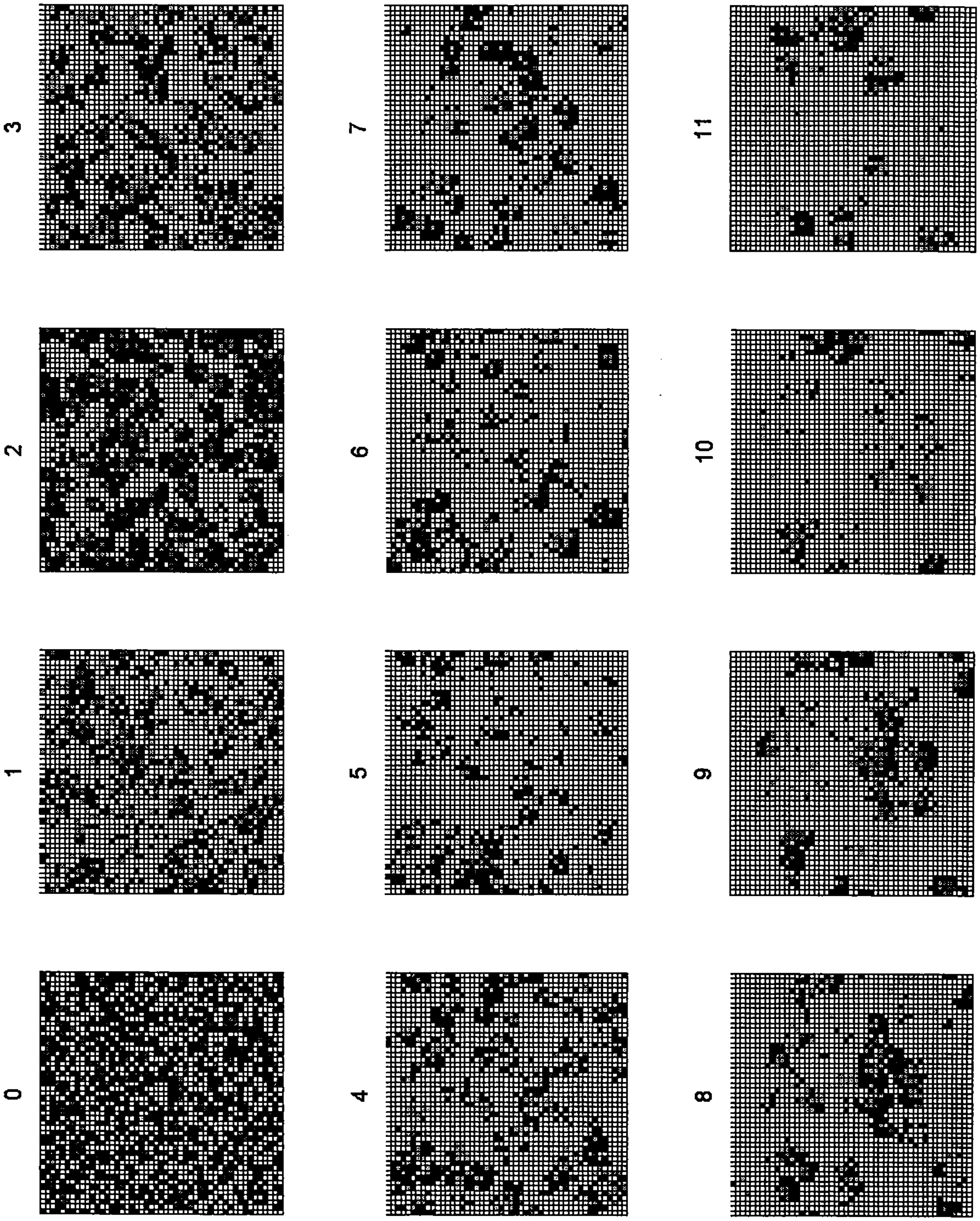


Figure #5.10(b) Relative abundance of a perennial herb species (#5166) in a grassland ecosystem with small mammalian herbivores.

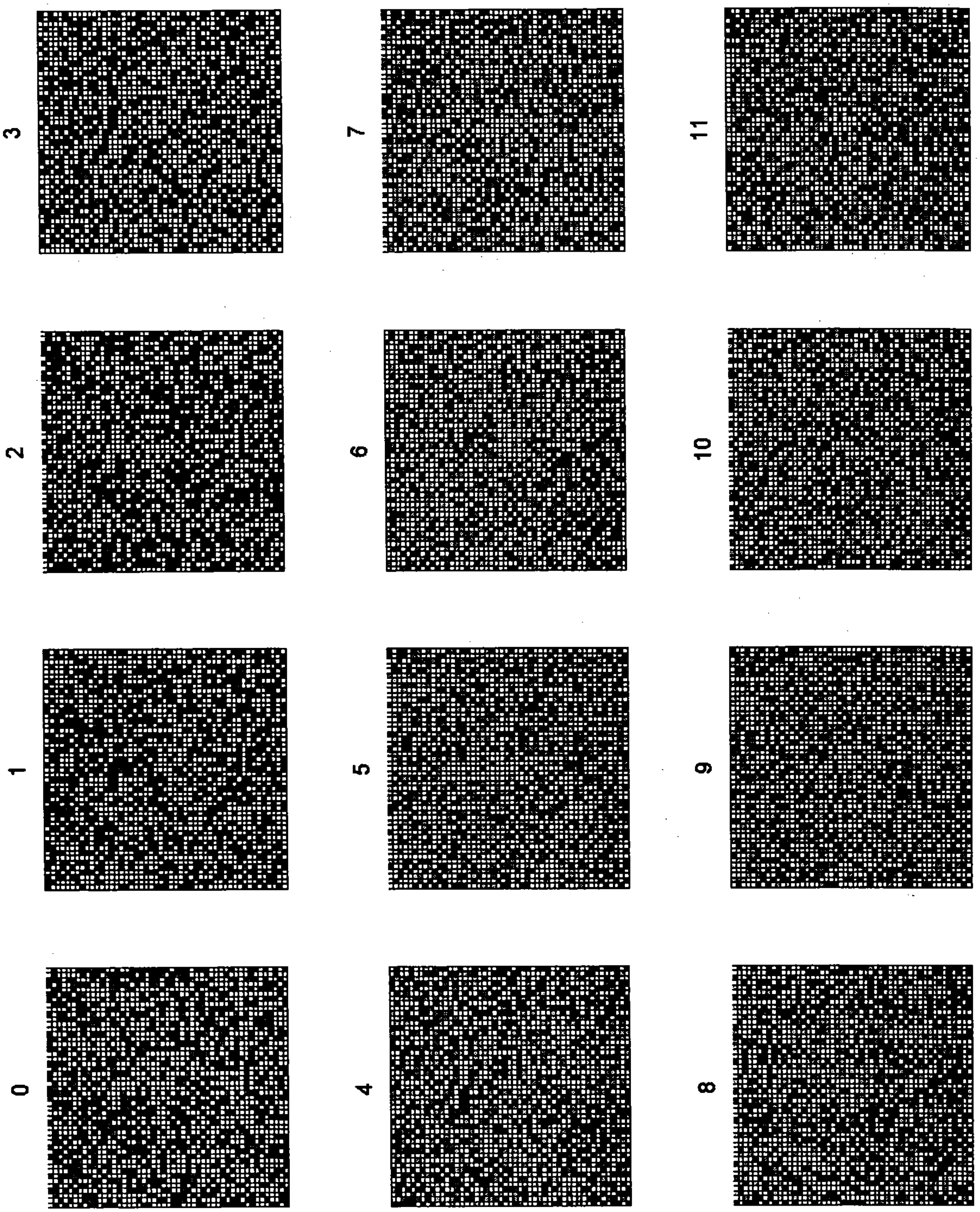


Figure #5.11(a): Relative abundance of a perennial grass species (#5146) in a grassland ecosystem, no herbivory.

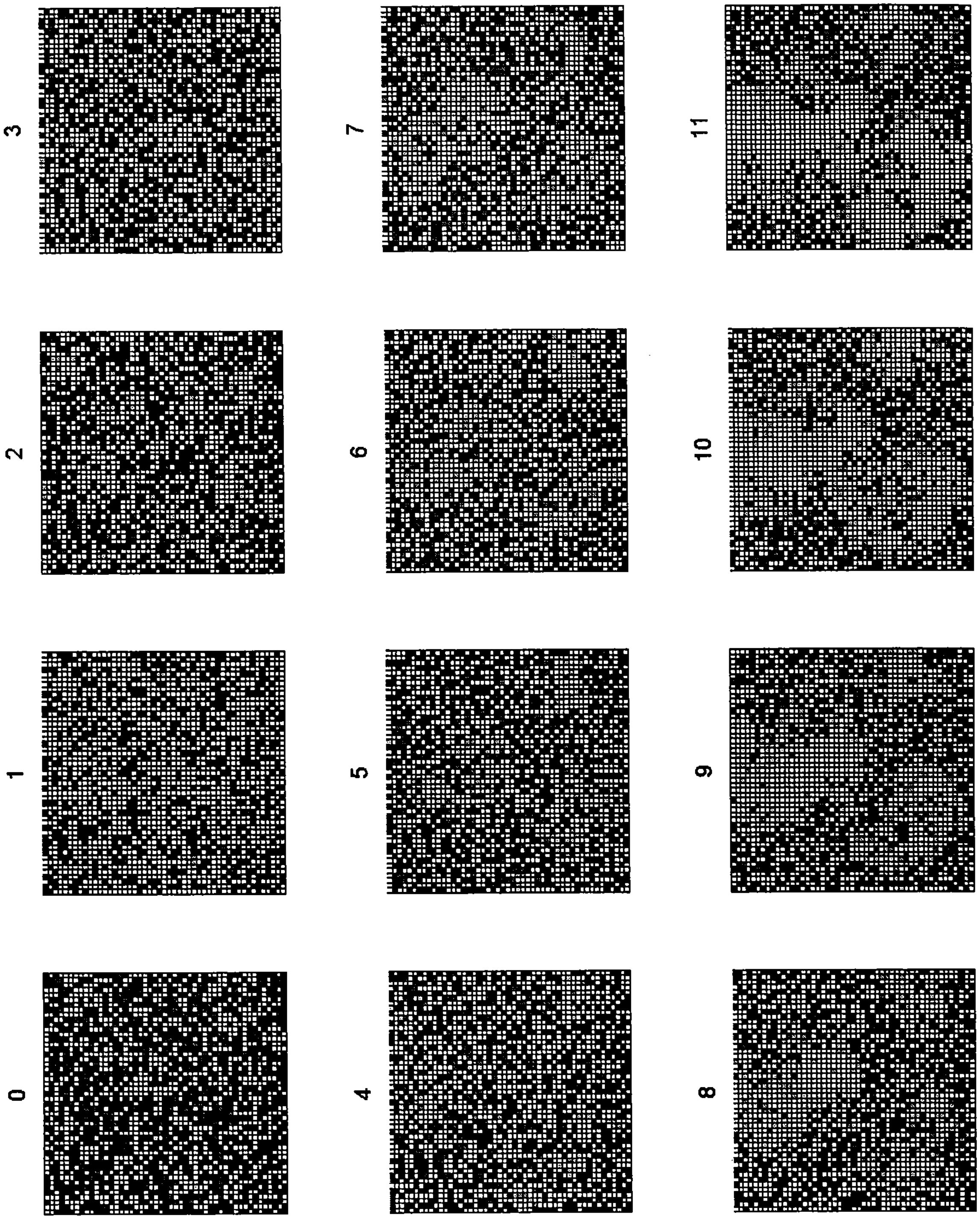


Figure #5.11(b) Relative abundance of a perennial grass species (#5146) in a grassland ecosystem with small mammalian herbivores.

(Figure 5.11), or the annual grass (Figure 5.9). Although this is not readily explainable, it is due to the different attribute values specified for the species, which determine their patterns of growth, and, therefore, their ability to compete with other plants for light and other resources, as well as to recover from the effects of predation by herbivores.

## 5.6. General discussion and conclusions

A generic, object-based primary producer model that can be configured to represent a variety of plant types was written and tested in the context of a larger ecosystem model and simulations based on that model. The approach, whereby each plant object represents an individual plant, or a small lump of plants, allows for the consideration and inclusion of a number of factors, such as competition for local resources and selective predation by grazers, that are not easily incorporated into crop or canopy models. The effects of these on plant population dynamics in multi-species ecosystem configurations were illustrated in several different simulations. In addition, the model predicted specific biomass levels that were similar to those of comparable, physical ecosystems.

The most significant parameters in the plant model (i.e., to which the performance of the model was most sensitive) are the photosynthesis and respiration rate constants, as well as those related to how the mass is apportioned within a plant (i.e., fractions of leaf, stem, root mass, etc.). These are all specified as species attributes, and the values assigned to them had the greatest influence on the maximum size (mass and total leaf area) reached by a plant, as well as on the length of time a plant took to reach maturity. Thus, in configuring different plant species for the ecosystem, special attention needs to be paid to these.

The overall ecosystem model is very computationally intensive, since it is based on an object-based approach whereby each object is updated during every time increment. Current simulations, based on models consisting of up to 50,000 plant objects (in addition to other ecosystem components such as terrain grid cells), pushed the envelope of our currently available hardware (450Mhz, Pentium III, 256 Mb of RAM; Windows 98). These proceeded at a pace of between 3-12 simulated years per day of real time and occupied most of the available memory. In the future, we hope to be able to perform longer simulations with larger numbers of objects, so that the model can be used to represent full-fledged forest ecosystems (which are composed of more tree objects than can currently be accommodated) as well as ecosystems covering larger terrains. Nonetheless, results from current simulations clearly illustrate the flexibility and effectiveness of such a high-resolution, object-based approach for the representation of different ecosystem configurations.

## 5.7. References

Acock, B., D. Charles-Edwards, D. Fitter, et al. 1978. The Contribution of Leaves from Different Levels within a Tomato Crop to Canopy Net Photosynthesis: An Experimental Examination of Two Canopy Models. *Journal of Experimental Biology* 29(111): 815-827.

Charles-Edwards, D., D. Doley, and G.M. Rimmington. 1986. *Modelling Plant Growth and Development*. Sydney: Academic Press.

Chave, J. 1999. Study of structural, successional and spatial patterns in tropical rain forests using TROLL, a spatially explicit forest model. *Ecological Modelling* 124: 233-254.

Chertov, O., A. Komarov, and A. Tsiplianovsky. 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.

Clark, J.S. 1992. Relationships among individual plant growth and the dynamics of populations and ecosystems. In *Individual-Based Models and Approaches in Ecology*, eds. D. DeAngelis and L. Gross, 421-454. New York: Routledge, Chapman and Hall, Inc.

Clark, O.G. 1999. *The Characterization of Biosystems.* Ph.D. diss., Department of Agricultural and Biosystems Engineering, McGill University, Montreal.

Fitz, H.C., E.B. DeBellevue, R. Costanza, et al. 1996. Development of a general ecosystem model for a range of scales and ecosystems. *Ecological Modelling* 88: 263-295.

Friend, A.D. 1995. PGEN: an integrated model of leaf photosynthesis, transpiration, and conductance. *Ecological Modelling* 77: 233-255.

Friend, A.D, A.K. Stevens, R.G. Knox and M.G.R. Cannell. 1997. A process-based, terrestrial biosphere model of ecosystem dynamics (Hybrid v3.0).

*Ecological Modelling* 95: 249-287.

Hansen, G. and C. Jensen. 1977. Growth and maintenance respiration in whole plants, tops, and roots of Lolium multiflorum. *Physiol. Plant.* 39: 155-164.

Hogeweg, P. and B. Hesper. 1990. Individual-oriented modelling in ecology. *Mathematical and Computational Modelling* 13(6): 83-90.

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. New York: Routledge, Chapman and Hall, Inc.

Marshall, B. and P.V. Biscoe. 1980. A Model for C3 Leaves Describing the Dependence of Net Photosynthesis on Irradiance. 1. Derivation. *Journal of Experimental Botany* 31(120): 29-39.

Parrott, L. and Kok, R. 2000a. Implementation of a generally configurable, object-based ecosystem model. Chapter 3 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and Kok, R. 2000b. Baseline performance of an ecosystem model and simulation for the EcoCyborg Project. Chapter 4 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Shugart, H.H., T.M. Smith and W.M. Post. 1992. The potential for application of individual-based simulation models for assessing the effects of global change. *Annual Review of Ecological Systems* 23: 15-38.

Smith, G., W. Stephens, P. Burgess et al. 1993. Effects of light, temperature, irrigation and fertilizer on photosynthetic rate in tea (*Camellia sinensis*). *Expl.*

*Agric.* 29: 291-306.

Thornley, J. 1998. Dynamic model of leaf photosynthesis with acclimation to light and nitrogen. *Annals of Botany* 81: 421-430.

Thornley, J. 1991. A transport-resistance model of forest growth and partitioning. *Annals of Botany* 68: 211-226.

Thornley, J. and I. Johnson. 1990. *Plant and Crop Modelling: A Mathematical Approach to Plant and Crop Physiology*. Oxford: Clarendon Press.

Thornley, J. 1976. *Mathematical Models in Plant Physiology.* London: Academic Press.

Whittaker, R.H. 1975. *Communities and Ecosystems,* 2nd Ed. New York: Macmillan Publishing Co., Inc.

### 5.8. Appendix: Set of attributes used to describe the state of a plant object

See next page.

## 5.8. Appendix I: Set of attributes used to describe the state of a plant object

| Attribute name | Description | Units |
|---|---|---|
| ID.NUMBER | instance identifier | - |
| STATUS | alive or dead | - |
| WHAT.ACT.METHOD | act method identifier | - |
| WHAT.GROW.METHOD | grow method identifier | - |
| ID.LEVEL1 | ancestor class | - |
| ID.LEVEL2 | ancestor class | - |
| ID.LEVEL3 | ancestor class | - |
| ID.LEVEL4 | ancestor class | - |
| ID.LEVEL5 | ancestor class | - |
| PLANT.DEVSTAGE | development stage | - |
| AGE | age | s |
| BIRTHDATE | "date" at which a new plant object (seed stage) is created | s |
| X.COORDINATE | x coordinate | - |
| Y.COORDINATE | y coordinate | - |
| IS.LUMP | true or false | boolean |
| NUM.UNITS | number of units in a lumped plant | - |
| MASS | total mass | kg |
| UNIT.MASS | mass of one unit in a lumped plant | kg |
| MASS.SEED | seed mass | kg |
| MASS.STEM | stem mass | kg |
| MASS.ROOT | root mass | kg |
| MASS.LEAF | leaf mass | kg |
| MASS.ORG.STOR | organic storage mass | kg |
| MASS.INORG.STOR | inorganic storage mass | kg |
| UNGERMSEEDMASS | mass of ungerminated seeds in lumped plant | kg |
| DRY.MASS | dry mass of plant | kg |
| MASS.NITROGEN | mass of elemental nitrogen in plant | kg |
| MASS.CARBON | mass of elemental carbon in plant | kg |
| MASS.OXYGEN | mass of elemental oxygen in plant | kg |

| | | |
|---|---|---|
| MASS.HYDROGEN | mass of elemental hydrogen in plant | kg |
| MASS.CARBO | mass of carbohydrate in plant | kg |
| MASS.FAT | mass of fat in plant | kg |
| MASS.PROTEIN | mass of protein in plant | kg |
| MASS.WATER | mass of water in plant | kg |
| MASS.INORGN | mass of inorganic nitrogen in plant | kg |
| SEED.CARBO | mass of carbohydrate in seed | kg |
| SEED.WATER | mass of water in seed | kg |
| SEED.PROTEIN | mass of protein in seed | kg |
| ORG.STOR.CARBO | mass of carbohydrate in organic storage | kg |
| ORG.STOR.WATER | mass of water in organic storage | kg |
| ORG.STOR.PROT | mass of protein in organic storage | kg |
| STEM.CARBO | mass of carbohydrate in stem | kg |
| STEM.WATER | mass of water in stem | kg |
| STEM.PROTEIN | mass of protein in stem | kg |
| ROOT.CARBO | mass of carbohydrate in root | kg |
| ROOT.WATER | mass of water in root | kg |
| ROOT.PROTEIN | mass of protein in root | kg |
| LEAF.CARBO | mass of carbohydrate in leaf | kg |
| LEAF.WATER | mass of water in leaf | kg |
| LEAF.PROTEIN | mass of protein in leaf | kg |
| OPT.CARBO | optimum carbohydrate content of plant parts | - |
| OPT.WATER | optimum water content of plant parts | - |
| OPT.PROTEIN | optimum protein content of plant parts | - |
| CARBOASSIMILATE | mass of carbohydrate assimilate | kg |
| PROTASSSIMILATE | mass of protein assimilate | kg |
| MAX.ROOT.DEPTH | maximum depth of plant roots | m |
| MAX.WIDTH | maximum width of plant | m |
| LUMP.DIAMETER | diameter of area covered by lumped plant | m |
| UNIT.AREA | area covered by one unit in a lump | m2 |
| TOTAL.LEAF.AREA | total area of leaf on plant | m2 |
| LEAF.THICKNESS | thickness of a leaf | m |
| CANOPY | canopy level | - |

| | | |
|---|---|---|
| PHOTO.RATE | photosynthesis rate | kgO2·m2leaf·s-1 |
| MAX.PHOTO.RATE | maximum photosynthesis rate | kgCO2·m2leaf·s-1 |
| RESP.RATE | respiration rate | kgCO2·s-1 |
| LEAF.EFF | leaf efficiency (photosynthesis constant & resp. constant, c1) | kgCO2·J-1 |
| MAINT.COEFF | maintenance coefficient (respiration constant, c2) | - |
| THETA | (photosynthesis constant) | - |
| NABSORPTIONRATE | rate of inorganic nitrogen absorption by plant roots | kgN·kgroot·s-1 |
| H2OABSORPTRATE | maximum water absorption rate by plant roots | kgH2O·kgroot·s-1 |
| TRANSRATE | transpiration rate | kgH2O·m2leaf·s-1 |
| MAXTRANSRATE | maximum transpiration rate | kgH2O·m2leaf·s-1 |
| WATER.STRESS | degree of water stress (range 0-1) | - |
| WATER.TOLERANCE | lowest tolerated water content (fraction) | - |
| LITTERPRODRATE | rate of limb and leaf loss | kg·s-1 |
| ROOTLOSSRATE | rate of root turnover | kg·s-1 |
| LEAF.SEN.RATE | rate of seasonal leaf senescence | kgN·s-1 |
| STORAGEUSERATE | rate of mass transfer from organic storage to new growth | kg·s-1 |
| FSGLEAF | mass fraction of growing seedling that is leaf | - |
| FSGSTEM | mass fraction of growing seedling that is stem | - |
| FSGROOT | mass fraction of growing seedling that is root | - |
| FSGSEED | mass fraction of growing seedling that is seed | - |
| FSGORGSTOR | mass fraction of growing seedling that is organic storage | - |
| FSGINORGSTOR | mass fraction of growing seedling that is inorganic storage | - |
| FGLEAF | mass fraction of growing mature or senescing plant that is leaf | - |
| FGSTEM | mass fraction of growing mature or senescing plant that is stem | - |
| FGROOT | mass fraction of growing mature or senescing plant that is root | - |
| FGSEED | mass fraction of growing mature or senescing plant that is seed | - |
| FGORGSTOR | mass fraction of growing mature or senescing plant that is organic st | - |
| FGINORGSTOR | mass fraction of growing mature or senescing plant that is inorganic | - |
| FRLEAF | mass fraction of reproducing mature plant that is leaf | - |
| FRSTEM | mass fraction of reproducing mature plant that is stem | - |
| FRROOT | mass fraction of reproducing mature plant that is root | - |
| FRSEED | mass fraction of reproducing mature plant that is seed | - |
| FRORGSTOR | mass fraction of reproducing mature plant that is organic storage | - |

| | | |
|---|---|---|
| FRINORGSTOR | mass fraction of reproducing mature plant that is inorganic storage | - |
| FDLEAF | mass fraction of dormant plant that is leaf | - |
| FDSTEM | mass fraction of dormant plant that is stem | - |
| FDROOT | mass fraction of dormant plant that is root | - |
| FDSEED | mass fraction of dormant plant that is seed | - |
| FDORGSTOR | mass fraction of dormant plant that is organic storage | - |
| FDINORGSTOR | mass fraction of dormant plant that is inorganic storage | - |
| PERENNIAL | true or false | boolean |
| HAS.LEAVES | true or false | boolean |
| HAS.SEEDS | true or false | boolean |
| IS.N.FIXER | true or false | boolean |
| LATENT | true or false | boolean |
| MIN.TEMP.GROW | minimum temperature for growth/photosynthesis | ˚C |
| MAX.TEMP.GROW | maximum temperature for growth/photosynthesis | ˚C |
| MAX.MASS.PLANT | maximum total mass | kg |
| MAXIMUM.AGE | maximum age | s |
| SEED.LIFE | maximum viable life of an ungerminated seed | s |
| GERMINATION | "date" at which a seed germinates | s |
| GERM.HOUR.LIGHT | number of daylight hours required by a seed to germinate | - |
| SEEDLING.AGE | age at which plant shifts from the seedling to mature growth phase | s |
| SEEDING.AGE | age at which seeds are produced | s |
| GROW.START | time of year that vegetative growth begins | s |
| GROW.END | time of year that the dormant period begins | s |
| S.SEED.MASS | mass of a single seed | kg |
| START.SEEDING | time of year that seeds begin to be formed | s |
| SEEDING.TIME | length of time for seeds to develop on plant | s |
| SEEDSURVIVERATE | fraction of dispersed seeds that are given the opportunity to germina | - |
| SEED.SIZE.CLASS | type of seeds produced by this plant | - |

## *Connecting text between Chapters 5 and 6*

Chapters 5 and 6 are complementary to one another in that they both contain descriptions of the biological component realm of the ecosystem model. Chapter 6 is a continuation of the discussion begun in Chapter 5, and consists of a detailed description of the manner in which higher trophic levels (animals) are represented. Further simulation results are presented, in which the impact of grazers on plant life, and the population dynamics that ensue, is explored in more detail. Lastly, in the Appendix of Chapter 6, the results of a long-term multi-trophic level simulation are compared with those of the Baseline simulations presented in Chapter 4.

The animal methods discussed in Chapter 6 are encoded in the file "bio.c", available on the enclosed CD-ROM (Appendix B).

Chapter 6 will serve as the working text for an article being coauthored by L. Parrott and R. Kok.

# Chapter 6. Higher trophic levels in a generally configurable ecosystem model

## Abstract

In this article, a description is given of the manner in which higher trophic levels (animals) are represented in a generally configurable ecosystem model. The animals are represented with a hybrid individual/lump-based method. Their behaviours and physiological functions are described with simple, mechanistic rules that are derived from various assumptions about growth rates, metabolic requirements, digestion and assimilation of food, gestation, etc. The animals interact in a detailed, spatially explicit environment that consists of a terrain, an atmosphere, and primary producers. The model has been implemented in simulation to explore population dynamics in multi-species ecosystems configured with two and three trophic levels. Sample simulation results are presented, together with a discussion of the effectiveness of the approach for the representation of animals in ecosystem modelling.

## 6.1. Introduction

An animal model has been written as one part of a larger model in which most of the major aspects of an ecosystem - both abiotic and biotic - are represented in an object-based manner. The larger model has been written as an engineering tool with which to study the design of ecosystems that have been cyborged with intelligent control systems. These systems are called *ecocyborgs*. The current model was written and configured to represent a hypothetical setting of an ecosystem situated in a relatively small, materially-closed space station. Although it will ultimately be linked in simulation with a sophisticated control system, the model is currently being implemented with only a few rudimentary control components. In this article, the animal portion of the model is described in detail; further descriptions of the model setting and other aspects were given in Parrott and Kok (2000b; 2000c).

## 6.2. Literature review

Although it is generally recognized that the comportment of an ecosystem is the aggregate result of many interacting abiotic and biotic components, ecosystems have conventionally been modelled with sets of equations that describe global level processes as continuous functions of mass and energy flows. Recently, however, the focus of ecosystem modelling has shifted towards the increasing use of object-based approaches, in which components in a system are represented as discrete entities whose functioning and relationships with other objects are described with sets of simple rules (DeAngelis and Gross, 1992; Hogeweg and Hesper, 1990; Huston et al. 1988; Judson, 1994). Although ecosystems may be represented at any level of resolution with this approach, usually, this is done at a component level, where, in the majority of cases, the objects correspond either to small-scale spatial patches (often a grid-based, cellular automata model is used for this) or individual organisms, and sometimes a combination of both (see Lett et al. (1999) for a comparison of the two types). This allows for the study of how, and to what extent, lower level mechanisms contribute to global level dynamics in ecosystems. Many object-based ecosystem models have been developed, and the approach has recently been increasing in popularity due to both its conceptual appeal, and the fact that it facilitates the study of certain emergent features in ecosystems that arise as the result of ecological self-organisation (Fahse et al., 1998).

In the vast majority of object-based ecosystem models, plants, but not animals, are included in the biological community. This is probably due to two main factors: (1) a perception that the global impact of animal behaviour has a negligible effect on overall system dynamics (e.g., in models written to assess global climate change (Friend et al., 1997; Shugart et al., 1992), only vegetation cover is included in representations of the Earth's biosphere), and (2) the inclusion of animals greatly increases the computational requirements of any model when implemented in simulation, due to the number of behavioural and/or physiological functions that need to be incorporated. Object-based ecosystem models in which animals are included tend to comprise very limited representations of the environment, in which vegetation is usually not explicitly present. There are, for example, many models in which animals are represented as simple entities that move about a spatially-explicit world, eating from lumped food sources (Beecham and Farnsworth, 1998; Booth, 1997).

When animals are included in object-based models, since they are mobile, they are usually represented as individuals rather than as patches of a cellular automaton (the ant colony model of Miramontes et al. (1993) is exceptional in this regard). In the vast majority of these models, the representations of animals are very simple, with a focus on behavioural, rather than physiological functions. Such models are generally written to explore possible mechanisms that may give rise to particular features of animal communities, such as patterns of distribution, or social structures. In the model described by Letcher et al. (1998), for example, the territorial behaviour of individual woodpeckers is described with simple rules in order to explore the distribution of the birds on a wooded landscape. Another example is the individual-based model described in Henein et al. (1998), which was written to explore the effects of landscape heterogeneity and disturbance on the population size and persistence of eastern chipmunks and white footed mice. In this model, the state of each animal was represented with only a few variables, and its behaviour was described with a function that determined the probability that it would die, reproduce, or move in a random direction during a given time increment. The probability of each of these three possibilities was different for the two species, and was dependent on the type of landscape patch in which the animal was situated. This model was used to study the relative success of the two species for different types of landscape structure and connectivity. A similar model was written by Gustafson and Gardner (1996) to explore the effects of heterogeneity in the landscape on the dispersal patterns of organisms. Animals were represented as "random walkers" that moved about the landscape according to a simple rule set.

An alternative to the ecologically oriented individual-based models are the agent-based models, in which a genotype/phenotype relationship is included. In these models, animals are represented as fairly abstract entities (i.e., unlike animals in individual-based models, these bear little resemblance to biological organisms) that evolve, or adapt their genotypic information as a result of their interactions with one another and the environment. Many of these models are based on ECHO, a program in which a genetic algorithm is used to emulate the evolution of organism-like "agents" in an artificial world (Holland, 1975). Hraber and Milne (1997), for example, have used a variation of ECHO to explore the effects of genome-mediated interactions on the persistence of a species. The RAM model (Taylor et al., 1988) is an exception amongst agent-based models, in which a more realistic representation of an animal's phenotypic and genotypic state is included in order to represent animal behaviour and learning strategies. RAM has been used to model lek formation by sage grouse, and to study the effectiveness of a mosquito control program in which insecticide was applied over a large area.

In none of the models described above are physiological considerations incorporated to elicit animal behaviour (even eating is usually regulated by a rule such as "eat once per day", rather than "eat when you are hungry"). The model presented in this article is unique in the sense that it introduces a mechanistic representation of animals, such that an animal's physiological state underlies much of its behavioural decision-making. This necessitates a considerably more detailed representation of animals than what is usually included in ecosystem models. In addition, whereas in all of the models described above, the simulation time increment is quite large (sometimes a year, or months) and fairly small population sizes are modelled, in the ecosystem model described here, large numbers of animals are modelled in a discrete manner, and a regular simulation time increment of 10 minutes is used.

## 6.3. Overview of the modelling approach

The objective when developing the model was to represent animals in a reasonably realistic, yet generic manner, with a focus on processes related to growth, metabolism and reproduction. These processes were modelled with an emphasis on material flows and subsequent mass distribution, conversion, accumulation, etc., in the animal body. Behavioural functions are also included, but these are currently fairly rudimentary, and there are no mechanisms to accommodate learning, adaptation, or genetic evolution. The model was written so that it can be applied to a wide variety of biological classes, including mammals, birds, reptiles and insects, with a minimum number of modifications.

Animals are modelled as objects that may represent either individuals, or "lumps" of individuals. The decision to use a hybrid of lumps and individuals was mostly based on the practical limitations of implementing a model in simulation with such large numbers of discrete objects. Lumps, therefore, are groups of organisms that function as single units. For the most part, animals belonging to mammal, bird, reptile and amphibian species are dealt with as individuals, and animals belonging to insect, arachnid, and mollusc species are regarded as lumps. In general, when the survival of an individual begins to have some consequence within the social order of its community, that species is dealt with on an individual, rather than a lumped basis. The death of a worker ant, for example, makes little difference to an ant colony, thus ants are modelled as lumps. The death of a female mammal caring for five young, however, will influence the survival of others, thus mammals are modelled as individuals. This is, by necessity, a rather arbitrary arrangement. In all, due to limitations of the available hardware, the current version of the model can accommodate up to 100,000 biological objects (plant or animal); even though many species are represented with collections of lumped objects, this limit is often approached in simulations.

Although the model is encoded in a procedural language (ANSI compliant C), all biological objects, including animals, are represented in a manner that embodies some of the features of object-oriented programming. The state of each object is described with a collection of "attribute" values[1], and its functionality is encoded in object "methods". Each object is an "instance" of a species class, which inherits some of its attributes and methods from a hierarchy of ancestor classes that are defined during the model configuration procedure. All animals are descendents of a Heterotroph class, which is then subdivided into Carnivore, Herbivore, Omnivore and Detrivore classes, which are in turn subdivided into Mammal, Reptile and Bird classes, etc. Species classes are at the bottom of the hierarchy. Animals retain knowledge of their ancestry, which determines the manner in which they carry out some methods. The use of these object-oriented features allows for the accommodation of many types of animals, with maximum reuse of code. It also facilitates the definition of many different species. The implementation of this was described in greater detail by Parrott and Kok (2000a).

Presently, only mammalian species (as well as plants) have been fully implemented and tested in simulation. These were all modelled as individuals, and some of their methods are, of course, applicable only to mammals. The majority of methods, however, have been written such that they are applicable to all types of animals. Thus, although the model description given below may seem to have a "mammalian bias", most of the assumptions and the methods presented are typical of those that will be used for all types of animals.

## 6.4. Description of the animal's world

Animals in the ecosystem model "live" in a spatially explicit environment which consists of a terrain and an atmosphere. The terrain is modelled as a collection of rectangular cells arranged in a regular 2D lattice. In current simulations, the size of the terrain is 500m x 500m, with each grid cell being 10m x 10m. The grid cells have properties that correspond to their elevations, saturated and unsaturated water contents, and various soil features (e.g., masses of decomposing and

---

[1] A complete list of the set of attributes used to describe animal objects is given in the article's appendix.

organic matter, etc.). The topography of the terrain currently used is gently sloping, with a central pond. Plant objects are situated on the terrain, and grow in response to environmental inputs (the system is driven by three climate related forcing functions: temperature, radiation intensity and rainfall). Animals move about the terrain freely, and consume the plants, as well as other animals.

| **The Elements** | | **The Compounds** | |
|---|---|---|---|
| *Name* | *Symbol* | *Name* | *Composition* |
| Carbon | C | Carbohydrate | $C_6H_{12}O_6$ |
| Dirt | D | Carbon Dioxide | $CO_2$ |
| Hydrogen | H | Compound Dirt | $D_2$ |
| Nitrogen | N | Fat | $C_{57}H_{110}O_6$ |
| Oxygen | O | Inorganic Nitrogen | $N_2$ |
| | | Molecular Nitrogen | $N_2$ |
| | | Molecular Oxygen | $O_2$ |
| | | Protein | $C_4H_{12}O_4N2$ |
| | | Water | $H_2O$ |

Table 6.1: Compounds and elements used to make up materials in the modelled ecosystem.

The ecosystem is modelled as consisting of 5 elements, from which 9 compounds may be formed (Table 6.1). Plants are made up of water, carbohydrate and protein, and animals are made up of water, protein and fat. The atmosphere is made up of molecular nitrogen, molecular oxygen, carbon dioxide and water. The terrain is made up of compound dirt, inorganic nitrogen, and water, as well as fat, protein, and carbohydrate which is derived from biological sources. Mass is entirely conserved in the system, and, during a simulation, a complete mass accounting is done for each of the elements and compounds. Almost every exchange between two objects in the system involves a transfer of mass, which is always accounted for, no matter how small the amount.

## 6.5. Representation of animals

### *6.5.1. The animal life cycle*

An animal object exists in the ecosystem for a finite period of time. Like a real animal, it is born, it lives for a while, and then it dies. During its life, an animal passes through a number of development stages, the lengths of which are specified for each species, until it dies of old age or some other cause. When an animal is born, it is a juvenile. It then reaches a weaning age, at which point it leaves its mother. After this, it continues to exist as a juvenile until it reaches an age of maturity. A mature animal is capable of reproducing, until it reaches the end of its reproductive life. The animal will then continue to live until it reaches its maximum age, at which point it dies.

### *6.5.2. The animal body*

An animal's body is modelled as being composed of two parts: *lean mass* (made up of protein and water) and *fatty mass* (made up of fat and water) (Figure 6.1). An animal also has a stomach, but the mass of food therein is not considered to be part of the animal's body as such; food can be incorporated through digestion and absorption. The masses of each body part, including the food in the stomach, and their constituent compounds and elements, are tracked for each animal instance.

For each animal species, a set of attribute values is used to define optimal fractions that describe how much of the body mass should be apportioned to the lean and fatty mass parts. Optimal fractions of protein and water in the lean mass, and of fat and water in the fatty mass, are also assigned as species attribute values. (Sample values are given in Table 6.2.) Thus, the approach used is somewhat "teleonomic" in the sense that a guiding hand, or overriding supervisory principle, operates to maintain the appropriate optimal proportions of materials in an animal's body. The optimal body composition is described only in terms of fractions; target masses are not specified for any of the body

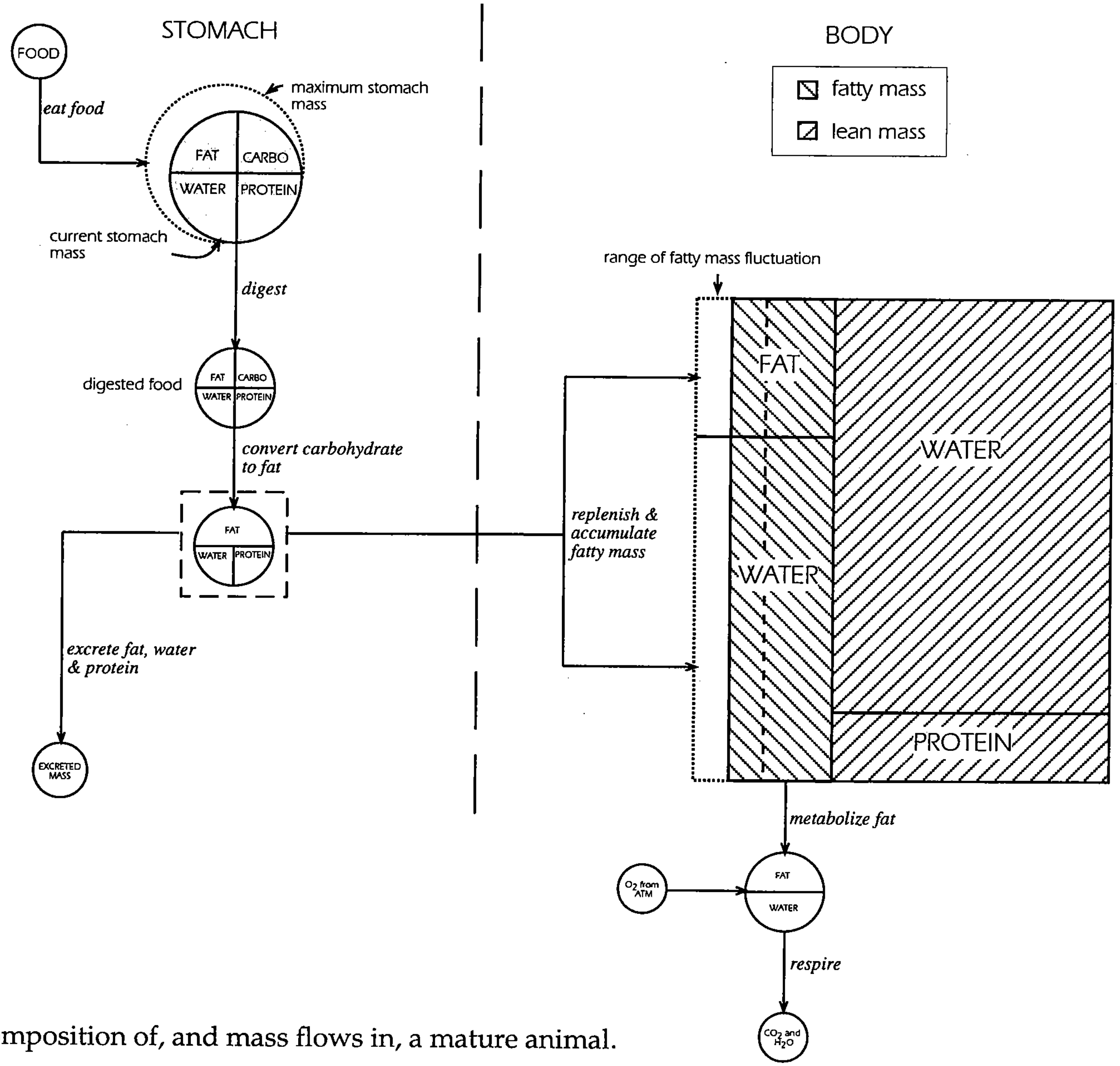


Figure 6.1: Composition of, and mass flows in, a mature animal.

parts. The mass of each body part, and of the animal as a whole, arises as a result of the growth function parameters specified for that species, in conjunction with the life history of the animal.

**Range of body composition values for modelled animals**

| | |
|---|---|
| Optimal lean mass fraction | 0.80-0.96 |
| Optimal fatty mass fraction | 0.04-0.20 |
| Maximum fatty mass fraction | 0.30-0.35 |
| Minimum fatty mass fraction | 0.03-0.07 |
| *Sub-fractions used for mammals* | |
| Optimal lean mass water content | 0.73 |
| Optimal lean mass protein content | 0.27 |
| Optimal fatty mass water content | 0.07 |
| Optimal fatty mass fat content | 0.93 |

Table 6.2: Values of body composition attributes for animal species. All values are expressed as fractions.

In a mature, non-gestating animal, the lean mass remains constant, and the fatty mass fluctuates, decreasing during periods of starvation, and increasing when food is plentiful. (Lean mass is never metabolized; an animal meets all of its energy requirements through the metabolism of fatty mass. See "Animal energetics" below.) For each species, values of minimum and maximum fatty mass fractions (relative to the total body mass) are specified. If an animal's body fat content decreases below the minimum, it dies of starvation. At the other extreme, fat accumulation slows down in animals with body fat contents that equal or exceed the maximum value. In a juvenile (growing) animal, the lean and fatty mass parts of the body increase in mass in proportion with the values specified for the optimal body fraction attributes (Figure 6.2). A gestating female

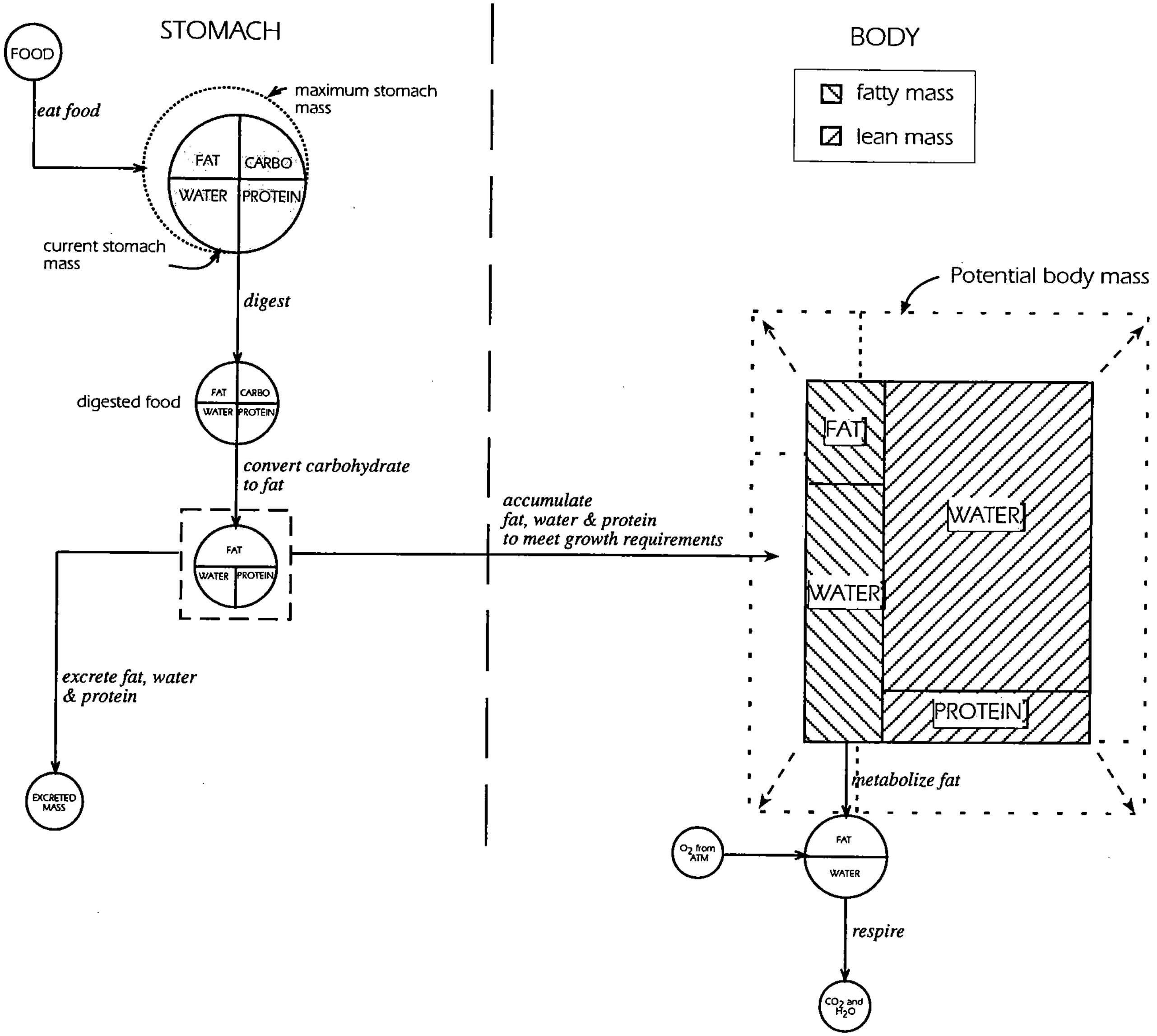


Figure 6.2: Composition of, and mass flows in, a juvenile animal.

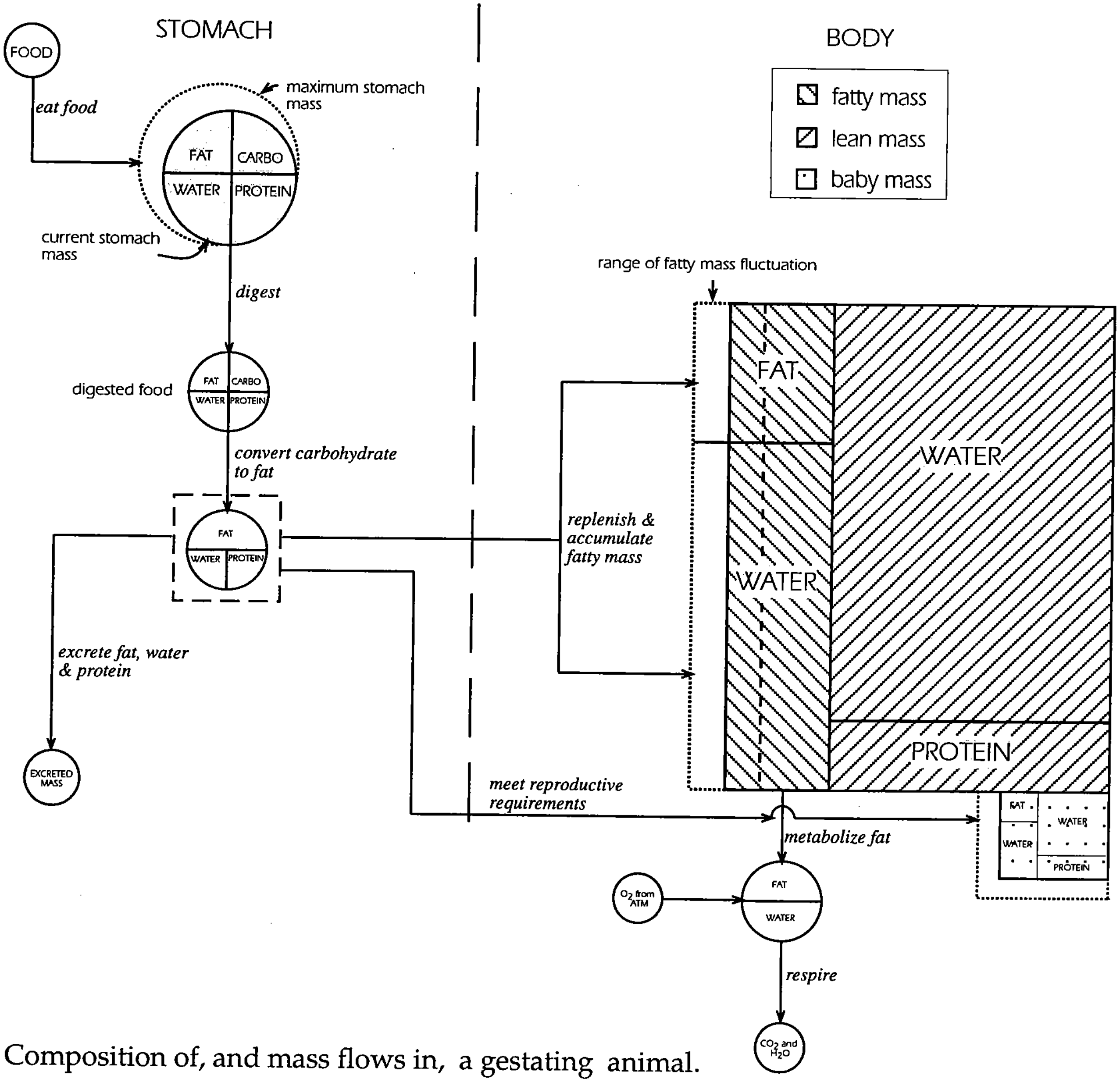


Figure 6.3: Composition of, and mass flows in, a gestating animal.

will also gain lean and fatty mass proportionally to support the development of a growing "baby mass" (Figure 6.3).

### 6.5.3. *Animal energetics*

A simple energy accounting scheme is used for animal objects, in order that reasonable estimates of respiration rates, and corresponding carbohydrate requirements, can be made based on body mass. Thus, for each time increment, an animal's total energy requirement (kcal/$\Delta$t)[1] is calculated by estimating how much energy is expended by the animal in order to carry out each function it performs, and adding this as follows:

$$\text{Total energy req'd} = \text{BMR} + \text{digestion} + \text{growth} + \text{reproduction} + \text{movement} \qquad (1)$$

where BMR stands for basal metabolic rate. This scheme does not currently include aspects related to thermoregulation. Thus, none of the costs of digestion, movement etc., are offset by internal heat generation, nor does the animal require additional energy to maintain (warm or cool) its body temperature.

During each time increment, the animal must metabolize a sufficient amount of fat to meet its total energy requirement. The metabolism of fat is modelled as a two-stage process, in which it is first converted to carbohydrate which is then metabolized via respiration. Stage 1, the conversion of fat to carbohydrate, is modelled as follows:

$$O_{2_{atm}} + H_2O_{body} + Fat_{body} \text{ -----> } CH_2O \qquad (2)$$

where oxygen from the atmosphere is combined with water and fat from the animal's fatty mass to produce carbohydrate. Next, in Stage 2, the carbohydrate is respired:

---

[1] Animal energetics was dealt with entirely in terms of kilocalories, rather than joules. This was done to accomodate the standard form of the basal metabolic rate equation that was used (Equation (4)). Rather than continually converting units, it made more sense to simply work with those that are most often reported in the literature on the subject.

$$CH_2O + O_{2_{atm}} \text{ -----> } CO_{2_{atm}} + H_2O_{atm} + \text{energy} \quad (3)$$

Thus, in this stage, oxygen is removed from the atmosphere, and carbon dioxide and water are expired to the atmosphere. Note also that Stage 1 has an associated respiration cost.

Minimally, an animal must have a sufficient amount of fatty mass to maintain its basal metabolic rate. If this is not the case, the animal dies. Any other activity can only be performed if sufficient energy is available. An animal's basal metabolic rate (BMR) is estimated as follows[1]:

$$Y = aX^b \quad (4)$$

where:

$Y$ = BMR ($\text{kcal} \cdot \text{day}^{-1}$)
$X$ = body mass (kg)
$a,b$ = empirical constants

This is a standard equation that has been widely used to model the basal metabolic rates of many different types of animals (Hill, 1976; Robbins, 1993), and whose form closely fits experimental data. In the ecosystem model, animal objects have attributes whose values correspond to the constants, a and b (these are species parameters). Robbins (1993, Table 8.3) has provided a summary of empirically derived values for various mammals, grouped according to body weight and food habits; these were used in the model for corresponding mammalian-type animals. Similarly, values of these constants for a range of bird and insect species have also been reported in the literature (Hill 1976; Swan, 1974). These will be used for birds and insects in future versions of the model. Although none of the animals hibernate as such, for some mammalian species, the BMR is decreased slightly during the winter.

---

[1] This equation has been kept in its non-metric form, since values reported in the literature for the constants a and b are usually derived for these units.

The energetic cost of growth or reproduction is assumed to be 0.25*BMR. The cost of digestion is assumed to be 30% of the energy in digested protein, 6% of that in carbohydrate and 13% of that in fat[1]. Thus, for every amount of protein, carbohydrate and fat that is digested, fat is metabolized to provide the required energy to "convert" the food. Similarly, other activities requiring movement (e.g., running or flying) have corresponding energetic costs. All of these are added together, as indicated in Equation (1) to calculate the total energy requirements of an animal during a given time increment. Each animal species has a maximum allowable respiration rate ($kgO_2 \cdot s^{-1}$), which cannot be exceeded and which therefore limits what animal instances can do (this has the effect of keeping an animal's activity at a reasonable level). For most mammalian species, the maximum respiration rate is set to ten times the BMR of a fully grown individual of average body mass.

### 6.5.4. Digestion

In the model, an animal must eat in order to meet its energy requirements, or else it will very quickly metabolize all of its fatty mass and ultimately die. When food is consumed by an animal, the mass that is consumed (which may be composed of any combination of carbohydrate, protein, fat, or water depending upon the food source) is placed in the animal's stomach. The stomach contents are then "digested" slowly over subsequent time increments. As food is digested, it is removed in small amounts from the stomach, and any carbohydrate is converted to fat, for assimilation into the body's fatty mass (Figure 6.1). The "digested" food is then available over the period of the time increment for assimilation, for purposes of growth or reproduction, or to be accumulated as fatty mass. At the end of a time increment, any remaining digested material is excreted by the animal and becomes part of the soil in that location.

---

[1] It is assumed that 5500 kcal (1314 kJ) can be obtained from 1 kg of protein, 4000 kcal (956 kJ) from 1 kg of carbohydrate, and 9000 kcal (2151 kJ) from 1 kg of fat.

### *6.5.5. Growth*

A juvenile animal's potential growth rate ($r_g$, kg·s$^{-1}$) at time, $t$ (s), is calculated as:

$$r_g = lm_t \cdot k_2 e^{-k_1 A} \cdot e^{k_3\left(\frac{lm_{p_t} - lm_t}{lm_{p_t}}\right)} \qquad (5)$$

where its potential lean mass ($lm_p$, kg) at age, $A$ (s) and time, $t$ is:

$$lm_{p_t} = lm_{p_{t-1}} + lm_{p_{t-1}} \cdot k_2 e^{-k_1 A} \cdot \Delta t \qquad (6)$$

and $k_1$, $k_2$, $k_3$ are species specific constants (stored as animal object attribute values). The second exponential term in Equation (5) is a "food availability" multiplier which causes the potential growth rate to be increased when an animal's lean mass is below the potential value for its age (see description below). An animal's potential lean mass increase during a given time increment is, therefore:

$$\Delta lm = r_g \cdot \Delta t \qquad (7)$$

The actual lean mass increase is a function of food availability. If there is a sufficient amount of digested food available, $\Delta lm$ (in the appropriate proportions of protein and water), plus an appropriate amount of fatty mass (fat and water) is assimilated into the body, and the animal has therefore grown. If there is not a sufficient amount of digested food to meet the potential growth rate, an amount less than $\Delta lm$ is assimilated. Thus, over time, if food resources are always low, a growing animal will end up being considerably underweight (as compared to what its mass would be were it to always achieve its potential growth rate). If, however, food resources are low for a brief period of time and then rebound, by virtue of the food availability multiplier, the growing animal is able to regain some of its lost potential lean mass. This allows an animal that has not

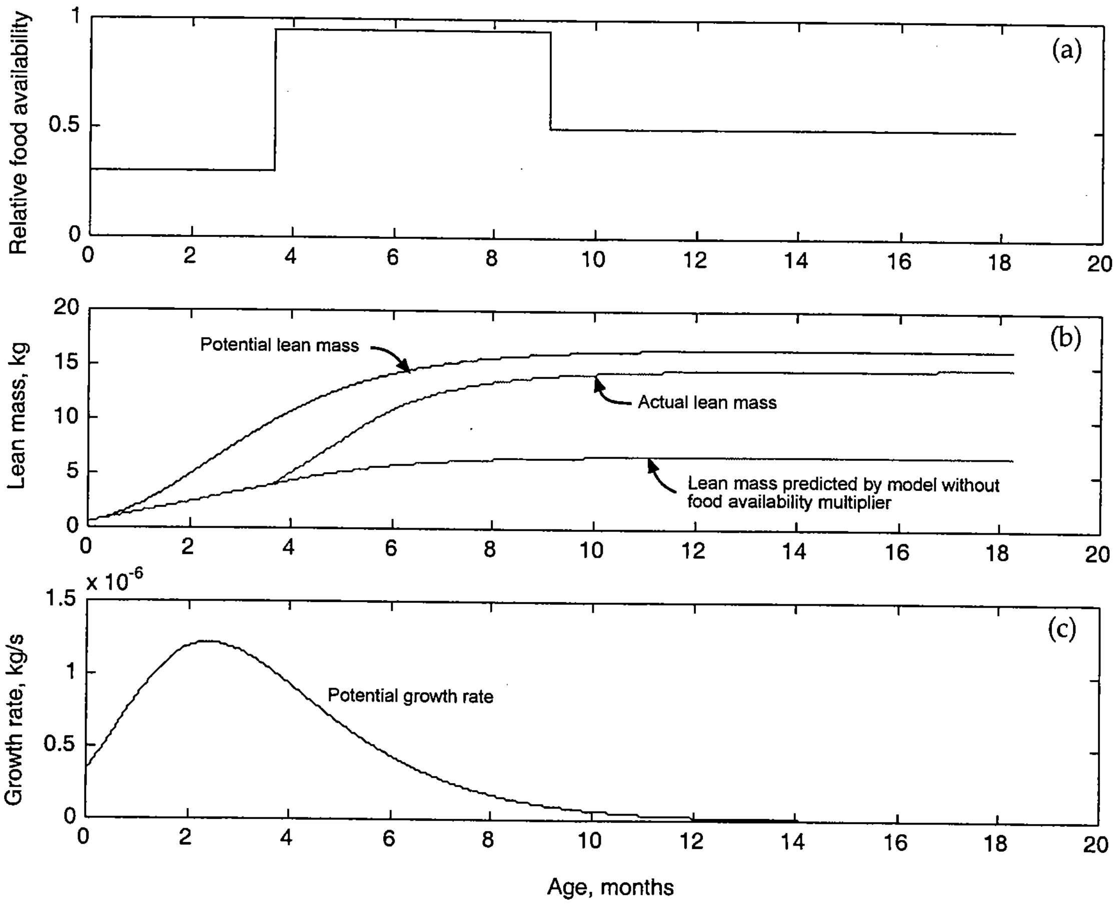


Figure 6.4: The animal growth function: (a) sample food availability scenario; (b) potential, actual, and non-adjusted lean mass growth curves given the food availability scenario shown in (a); (c) maximum possible lean mass growth rate. Constants used: $k_1$=2e-07; $k_2$=7e-07; $k_3$=4.0.

consistently met its growth expectations to "catch up" if food becomes more plentiful. The overall performance of the growth model for a hypothetical food availability scenario is illustrated in Figure 6.4.

According to the teleonomic scheme previously described, the body of an animal, mature or juvenile, must always be composed of optimal fractions of protein, fat and water. Thus, as an animal grows, all of its body parts (i.e., lean and fatty mass, and their constituents) must increase proportionally. When new mass is assimilated into the body from the digested food, there must be sufficient amounts of each required compound in order for the potential growth rate to be achieved. If there are any limiting compounds, the growth rate is correspondingly reduced.

### *6.5.6. Reproduction*

Reproduction is currently modelled after mammalian physiology; other types of reproduction will likely be implemented in future versions of the model, as bird, reptile, amphibian, and insect species are added. Thus, in the reproductive process described here, a mature female animal can become "pregnant". It then accumulates mass at a regular rate for the length of a "gestation period", after which the animal "gives birth" to one or more juvenile individuals of its species. All animal objects have a number of attributes related to reproduction. These include the length of the reproductive, or fertility, period (i.e., the time of year during which a female may become pregnant), the length of the gestation period, the size of a typical litter, etc. (see article appendix for a complete list of attributes).

Any female animal that has reached the age of maturity for its species and is neither currently pregnant nor with young, may become pregnant during the reproductive period. To do so, it must first find a suitable mate, i.e., a mature male of the same species, and move to the mate's location on the terrain. Once a mate is found, the female will then become pregnant according to a given probability of conception and must subsequently support the growth of a "baby mass" which is considered to be a temporary part of the female's body. During the gestation period, digested food is assimilated into the baby mass at a set baby growth rate (the value of which is a species attribute). The pregnancy is terminated early if, at any time, the female's ratio of fatty mass to body mass drops below a certain threshold, or if the accumulated baby mass is too low (usually this is due to poor maternal nutrition). When the female gives birth, one or more juvenile individuals are created, and their birth masses are removed from the baby mass in the mother's body. An animal will give birth to as many juveniles as specified by the value of its litter size attribute, so long as there is a sufficient amount of baby mass available. In cases of insufficient baby mass, as many as juveniles as possible will be created with the available baby mass, and in some cases a small excess will be removed from the mother's ordinary body mass. In this version of the model, juveniles are not fed by their mothers. They do, however, share the mother's home (see below) until they have reached a weaning age. A female with young cannot become pregnant.

### *6.5.7. Creation of new animal objects*

Each new animal object is initialised with a unique set of attribute values. Most of these are selected stochastically according to species specific distributions. For example, a juvenile's birth mass is selected randomly, according to a normal distribution that has a particular (species-specific) mean and standard deviation. To avoid extreme situations, this distribution is truncated at both ends according to species-specific minimum and maximum values. Other attribute values are computed similarly, sometimes according to other types of distributions (e.g., binomial or uniform). With this procedure, every juvenile is born unique, although representative of its species. The complete procedure for creating new instances, and for defining species attribute distributions, etc., is described in Parrott and Kok (2000a).

### *6.5.8. Consumer behaviour: Food searching and the eat preferences matrix*

An animal's food preferences are defined with a list of eat preference values that describe the relative delectability of each species. The eat preference values may range from zero to one, where a value of zero means that the animal would never eat a member of the target species, no matter what the circumstances. A value of one, on the other hand, means that the animal has a very high preference for the target species. A herbivore, for instance, may have eat preference values ranging between 0.5 and 0.9 for a variety of plant species, and have eat preference values of 0 for all animal species. Likewise, a carnivore would have positive eat preference values for a number of other animal species, but zero values for most plants. Several parts of the soil, such as seeds, decomposing plant and animal matter, and organic matter are also available for consumption, and each species is assigned corresponding eat preference values for these as well. Eat preferences are assigned for all species in the ecosystem during model configuration and these values are not modifiable during the course of a simulation.

When an animal is hungry, that is, when the amount of food in its stomach drops below a certain threshold value, it searches for food. To find food, the animal surveys its surroundings and selects the most desirable thing to eat based on a weighted consideration of both delectability and distance, computed as follows:

$$E = P * \frac{1}{e^{D/10}} \tag{8}$$

where,

E = food item edibility ($0<E\leq P$)
P = eat preference value for target food item ($0\leq P\leq 1$)
D = shortest distance between animal and food item (grid cell units)

Whether the animal actually attempts to eat the selected target is determined by its hunger level: If the animal is only marginally hungry, it will not eat a food for which it has an edibility value (E) that is less than a certain threshold (0.6 is usually used). As an animal's hunger level increases, however, it becomes less and less selective in its acceptance of food items. The maximum radial distance (from its current location) that each animal searches for food is determined by the value of a corresponding species attribute: small herbivores are limited to a distance of about 20m, whereas large carnivores might search the entire terrain. If a desirable food item is located and the decision is made to attempt to eat it, the animal will move toward it. If the food is reached, it will be eaten. An animal has no memory, thus if the food is not reached, the animal will not necessarily continue to move toward it in subsequent time steps (unless it searches for food again and selects the same food item as its target, which is, of course, a quite likely scenario). If a hungry animal does not find any appropriate food items during its initial survey of its surroundings, it will move in a random direction so as to increase its chances of finding food during the next time increment.

Although sometimes a food item is part of the soil, most often it is another living biological component object, which may be a plant or an animal. If the food item is part of the soil, the consumer will eat as much as it can, until either its stomach is full, or the food item is depleted. The removal of material from living food items, however, has somewhat greater implications. Plant objects can be partly eaten by a predator and remain alive afterward, whereas animals are always killed by a predator, and then consumed in part or in whole. A predator will eat as much of its prey (plant or animal) as it can hold in its stomach and the remains are left either as decomposing matter or as standing plant biomass, depending on the situation. In addition, the woody part of a plant is usually not consumed, and 15% of the mass of an animal is assumed to be non-consumable (to account for bones, fur, etc.). In the current version of the model, animal prey do not flee from approaching predators and are caught with a probability of 100%.

Thirst is dealt with similarly to hunger. If an animal's body water content falls below the optimal fraction, the animal is "thirsty" and will seek out water to drink. In this version of the model, there are no puddles of water on the terrain, thus all thirsty animals must travel to the central pond to drink.

### *6.5.9. Housing*

Most animals have a "home", which may be a permanent burrow or nest, or just a temporary resting place. Small herbivores, for example, have permanent homes to which they return to sleep at night. This has the effect of moderating the extent to which these animals move about the terrain, since they tend to stay reasonably close to their homes. Each animal has an attribute, the value of which describes whether or not it currently has a house. If this is true, the coordinates of the house on the terrain are also stored as attribute values. Females of all mammalian species establish homes for at least some period of time in order to give birth to, and shelter, their young until these have reached weaning ages. Juveniles leave their mother's home at this age and attempt to establish their own homes. Animals select home locations according to a crowd tolerance attribute whose value reflects the maximum number of animals of that species that is socially acceptable in a given grid cell. This may cause an animal to seek out uninhabited territory in order to establish a home.

### *6.5.10. Movement*

Each animal object has a known location on the terrain, recorded as a pair of coordinate attributes whose values correspond to x and y locations on the terrain grid. As an animal moves, its location in terms of grid cell coordinates is updated. An animal may leave its current location for various reasons, such as searching for food, or to reach a feature of the landscape such as the pond. Each animal object has a maximum travel speed which determines the total distance that it can move in one time increment. If an animal cannot reach a target location in one time increment, it will move part of the distance. During the next time increment, it may then continue to move in that direction, if it once again decides to perform that particular activity. Thus, it may take several iterations for a small animal object to reach a target location that is many grid cells away.

### *6.5.11. The Act method*

During a simulation, the states of the animal objects are updated in sequence, in the order they are stored in the population index vector (`gndx001`, see Parrott and Kok, 2000a; this is an unsorted list to the bottom of which new animal objects are added as they are created). When an animal instance is activated, the appropriate "Act" method is called. This method is called once per time increment for every animal in the ecosystem. There are several different Act methods, each of which is specific to a particular animal type, e.g., herbivorous mammal, carnivorous reptile, insect, etc. All animals of a similar type perform the same Act method. This allows for considerable differentiation between different types of animals in terms of their behavioural routines, without requiring the writing of an Act method for each species. In addition, the Act methods allow for a fair degree of variability in the behaviour of any particular animal, since the execution of many functions is conditional upon the values of the animal's attributes (i.e., its current state).

All Act methods share a common format, and are composed of four main sections: maintenance, mass assimilation, activity, and social behaviour, which are executed in that order. In the first two sections, essential body functions are executed; these must be carried out for each animal during every time increment. The activities in the third and fourth sections are, however, optional, and are only performed when conditions are appropriate (e.g., searching for food when hungry, or finding a mate during the mating season). An animal's default activity, should it not be hungry or needing to participate in any social activity, is "sleep".

In the maintenance section of the Act method, an animal's status (i.e., alive or dead) is verified, its age is increased, and its body composition is checked to ensure that it has a sufficient amount of fat, and is not lethally dehydrated. Next, the animal's current basal metabolic rate is calculated. In the mass assimilation section, food is digested from the animal's stomach, and is then partitioned and allocated as required to meet growth or reproductive requirements, and to replenish or build up the body's fatty mass. Next, the animal selects an appropriate activity to perform, depending upon its state. The activity may involve searching for and eating food, searching for a place to establish a home, or sleeping. Next, if the animal is not asleep, it participates in social behaviour, which presently only involves selecting a mate for reproductive purposes. Most animals return to their homes to sleep. Also, unless an animal's hunger level is below a certain threshold, it will sleep during the hours of darkness.

Sometimes an animal's behaviour affects the state of an object in the encompassment, or another biological component. The animal that is currently being treated is referred to as the "primary active instance". While it interacts with another biological component, that instance is referred to as the "secondary active instance". The attribute values of the secondary active instance may be updated as the result of activities of the primary active instance (e.g., if it is eaten, its status and mass attributes will need to be updated). If an animal interacts with the encompassment, through, for example, an exchange of mass with the atmosphere or terrain, the appropriate encompassment object is updated immediately.

## 6.6. Performance of the model

### *6.6.1. Testing and verification of individual animal objects*

As described above, the animal routines were written as part of a larger ecosystem model, the performance of which had been previously tested with only plant-based configurations. The animal routines were, therefore, tested and verified in the context of this larger model, which contains representations of an environment as well as other objects with which the animals must interact in order to survive. Animals belonging to herbivorous species were tested first. For each herbivorous species, only a few (<10) individuals were introduced to otherwise plant-based ecosystems. The behaviour and life history of each of these individuals was observed. The values of all of the individual's attributes were recorded for at least a few simulated years, and these were subsequently

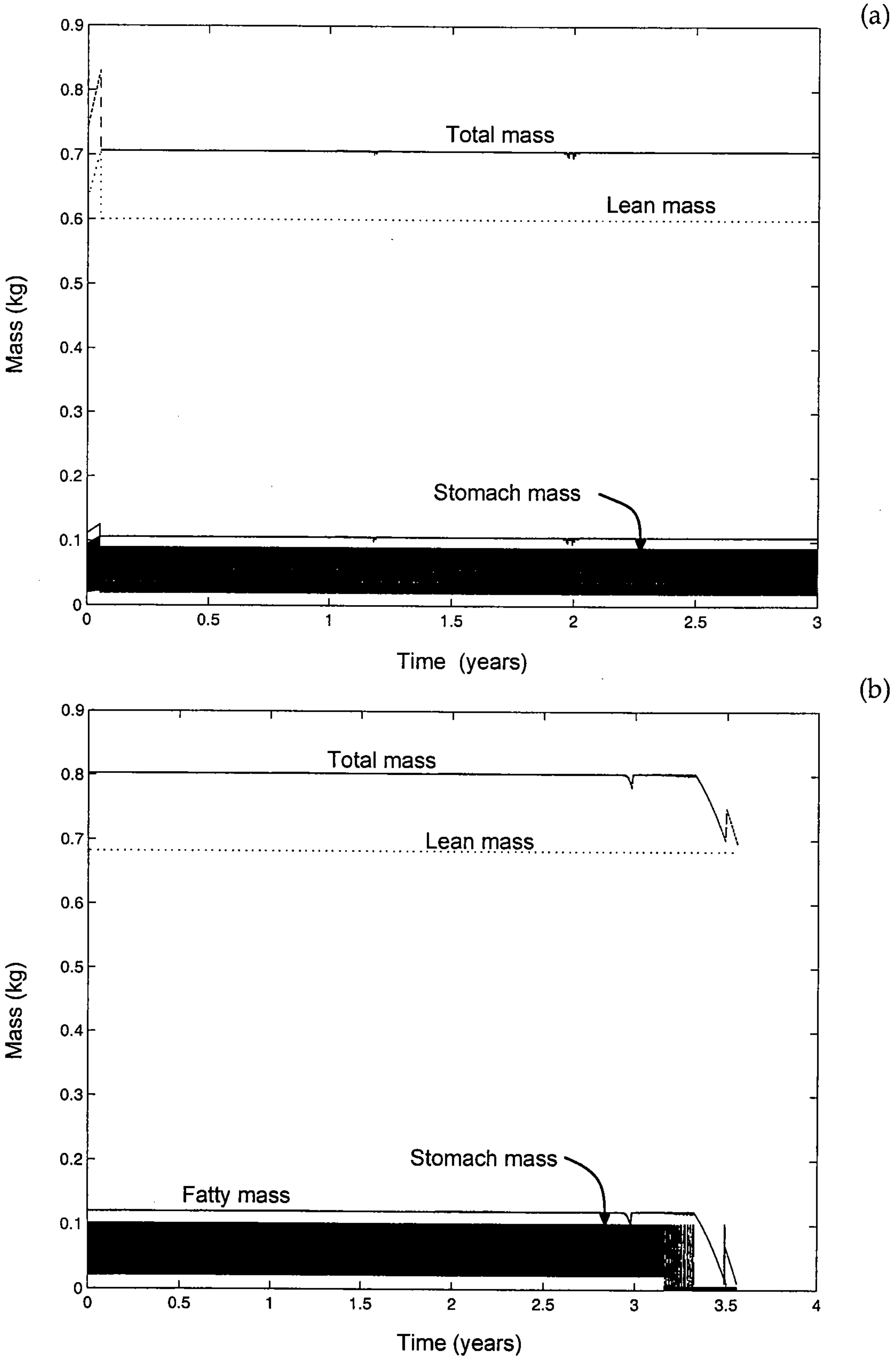


Figure 6.5: Partial life histories of two mature individuals used to initialize the system. (a) a female that starts out pregnant, gives birth, and then continues to live as usual; (b) an animal that eventually starves to death.

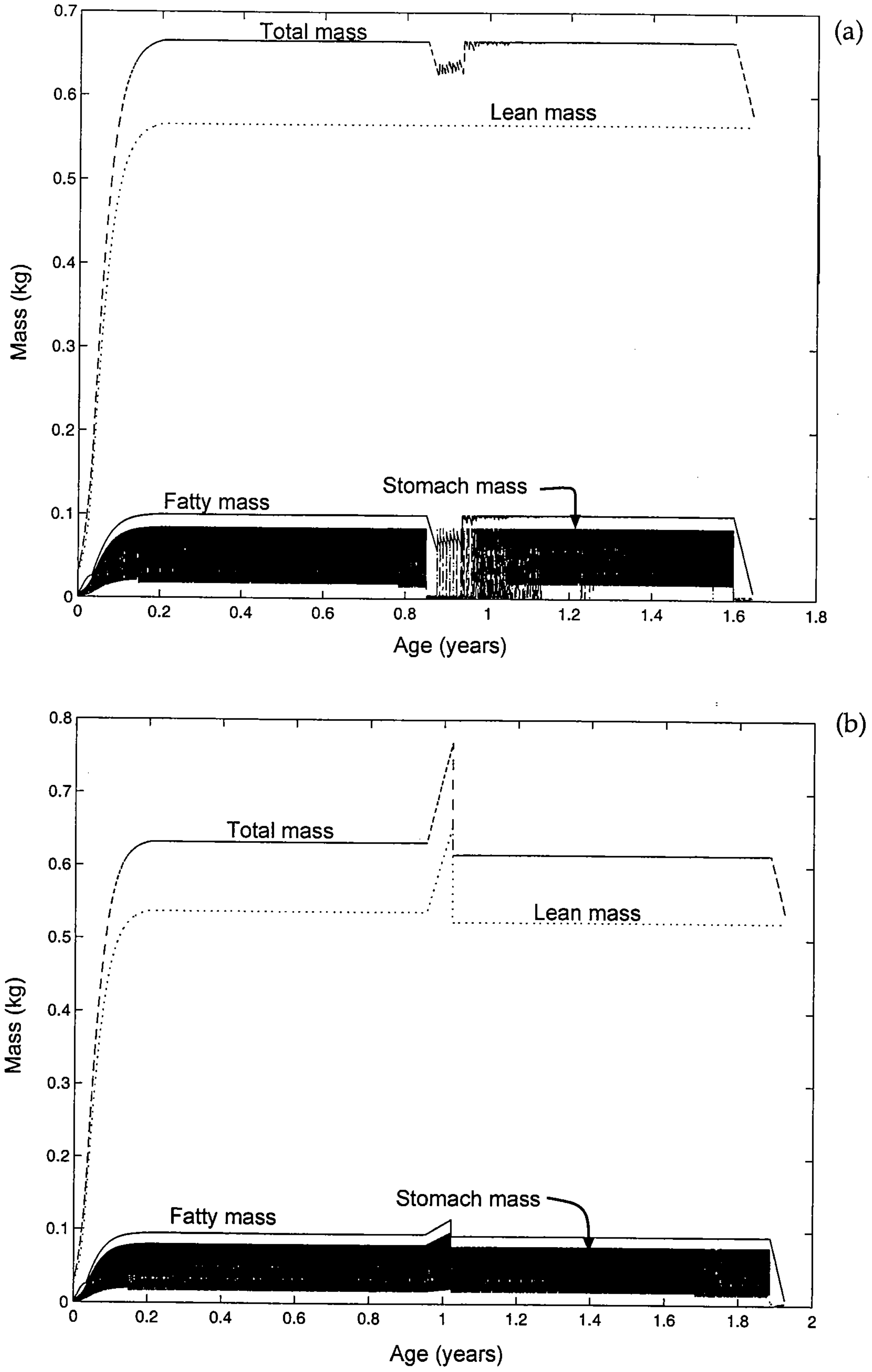


Figure 6.6: Life histories of two animals created during a simulation. (a) an individual that reaches maturity and then starves to death in its second year; (b) an individual that grows to maturity, undergoes a period of gestation, gives birth, and ultimately dies of starvation.

checked to ensure that they remained within expected ranges. The states of any new individuals created during such test simulations were also traced. Next, carnivorous species were tested in a similar manner, by adding a few individualsat a time to ecosystems configured with only plants and a few herbivores. Larger populations of animals were introduced to the ecosystem only once each species had been fully verified.

Overall, when implemented in simulation, the animal routines appear to successfully reproduce some aspects of animal, specifically mammalian, life forms, particularly with regards to their material and energetic requirements. Sample life histories of two mature individuals (created in mature states for the purpose of system initialisation) are shown in Figure 6.5. Similarly, the life histories of two individuals born during a simulation are shown in Figure 6.6. No attempt was made to emulate real organisms of any particular species with the sample individuals shown in these figures. Better species attribute values could be input to obtain results that more closely reflect the states of real organisms.

### *6.6.2. Multi-trophic level ecosystem simulations*

A number of simulations were run with multi-trophic level ecosystems in order to study the global population dynamics that arise due to the specification of various food webs. The results of four of these, AnimSim1, AnimSim2, AnimSim3 and AnimSim4 are reported here. For each of these simulations, the model was configured with the same abiotic environment,[1] but with different biological communities. The selected species and their initial population sizes for each of the four simulations are given in Table 6.3. For each simulation, the ecosystem configuration and the initial state were specified as described in Parrott and Kok (2000a), and each initial population was comprised of both mature and juvenile individuals. All four simulations were run for a period of 50

---

[1] Thesis note: The compositions and starting states of the encompassment and material storage realms were the same as those used for the baseline simulations (Chapter 4).

simulated years, on a Pentium III 450MHz computer with Windows 98 as the operating system. Simulation run-times were: AnimSim1, 18 days; AnimSim2, 12 days; AnimSim3, 3 days, and AnimSim4, 32 days. The resultant population dynamics for the four cases are shown in Figures 6.7, 6.8, 6.9 and 6.10.

| | AnimSim1 | AnimSim2 & AnimSim4 | AnimSim3 |
|---|---|---|---|
| **Animals** | # of instances | # of instances | # of instances |
| Species #5001: Small herbivore | 100 | 501 | 500 |
| Species #5005: Small herbivore | 103 | 2001 | 2001 |
| Species #5011: Gramnivore | 101 | 501 | 500 |
| Species #5322: Small carnivore | 0 | 0 | 1 |
| **Plants** | # of instances | # of instances | # of instances |
| Species #5068: Coniferous tree | 500 | 400 | 400 |
| Species #5076: Deciduous tree | 0 | 400 | 400 |
| Species #5144: Annual grass | 1200 | 2000 | 2000 |
| Species #5146: Annual grass | 1200 | 2000 | 2000 |
| Species #5166: Annual forb | 1200 | 2000 | 2000 |
| Species #5320: Perennial grass | 1200 | 2000 | 2000 |
| Species #5321: Perennial grass | 1200 | 2000 | 2000 |

Table 6.3. Selected species and initial population sizes for AnimSim1, 2, 3 & 4

As is evident from Table 6.3, for the first two simulations, the ecosystem was configured with two-trophic levels: plants and herbivores. The population dynamics that resulted in these two simulations were quite different: In AnimSim1 (Figure 6.7), all of the plants, except for the tree species, underwent an initial period of stress due to predation and competition with other plants, with only two species remaining after eight years. While the mostly gramnivorous species (#5011 which feeds on the seeds of species #5068) went extinct in the early years of the simulation, the other two herbivorous species (#5005 and #5001) remained intact. These fed upon the remaining two grass species and an interesting, cyclical predator-prey pattern emerged in the ensuing population dynamics (there were also yearly cycles, due to the annual growth and winter dormancy patterns of the plant species, which affects the populations that feed upon them). In contrast, in AnimSim2 (Figure 6.8), the ecosystem was also initialized with both plants and herbivores, but with quite different end results. In this simulation, the initial population sizes of the herbivores were considerably larger than those in AnimSim1, and it appears that they were too large to be supported by the available primary production. By year three, the population of animal species #5005 reached almost 15000 instances and the population of species #5001 was over 6000. During the following year, almost all of the herbaceous plants were consumed, and the animal populations severely crashed shortly thereafter. One grass species (#5146), however, recovered and re-established itself over most of the terrain. Both tree species survived, with species #5076 gaining dominance. Although the herbivorous species did not go extinct, their populations never recovered. Their population sizes at the end of the 50-year simulation period were: species #5001, 3 instances; species #5005, 20 instances and species #5011, 1 instance.

For AnimSim3 (Figure 6.9), the ecosystem was initialized with the same starting state as AnimSim2, except for the addition of a single instance of a carnivorous predator. The predator was added in an attempt to maintain the population sizes of the herbivores at levels that could persist without decimating the plant life. Although the predator had fairly moderate feeding requirements, its consumption level could not be supported by the available numbers of prey. The result was that the populations of herbivores were kept at fairly low levels, but the combination of predation by the carnivore with low primary production by the over-consumed plants led to the extinction of all of the herbivorous species in a few years. The single carnivore (not shown in Figure 6.9) died of starvation several weeks after all of the herbivores were gone. The addition of the carnivore did, however, substantially affect the end state of the system as

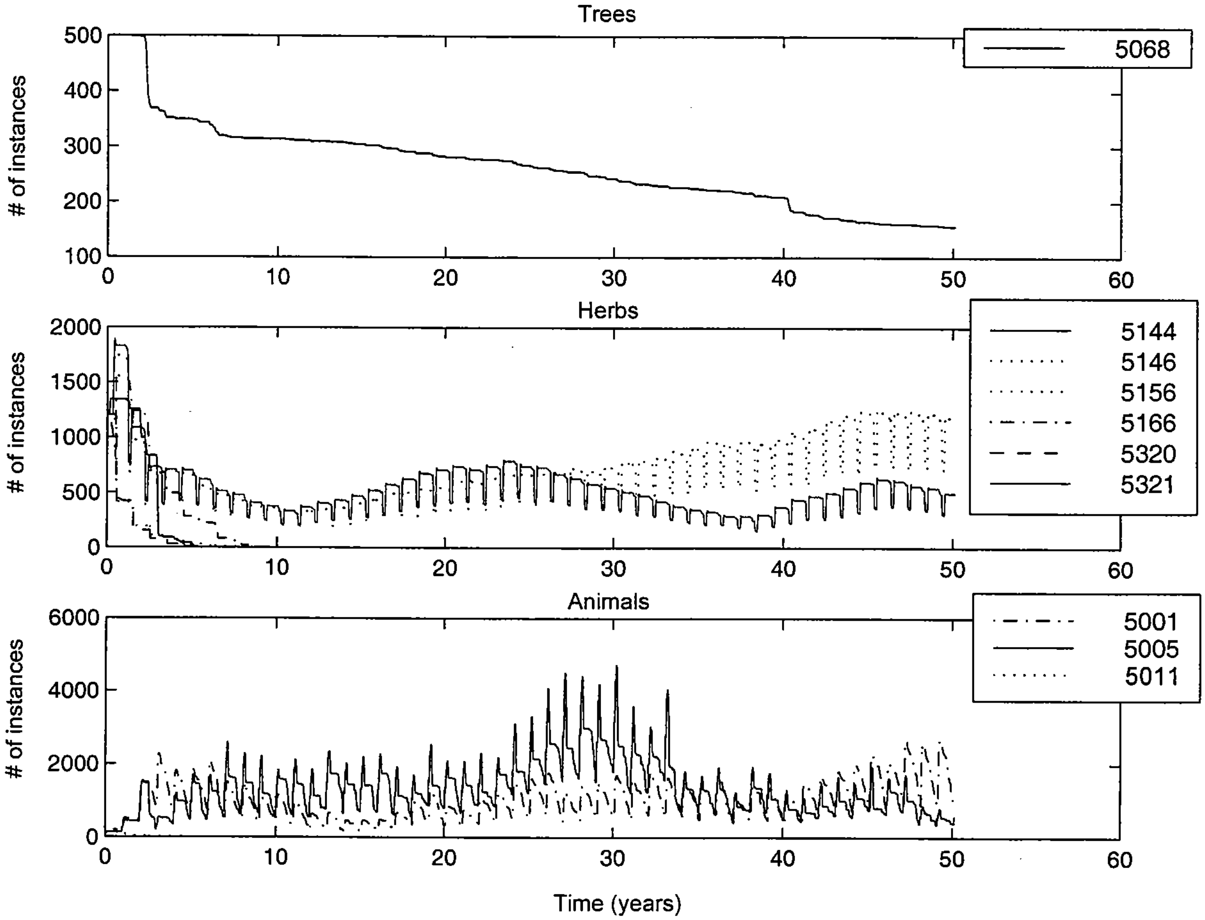


Figure 6.7: Population dynamics for AnimSim1.

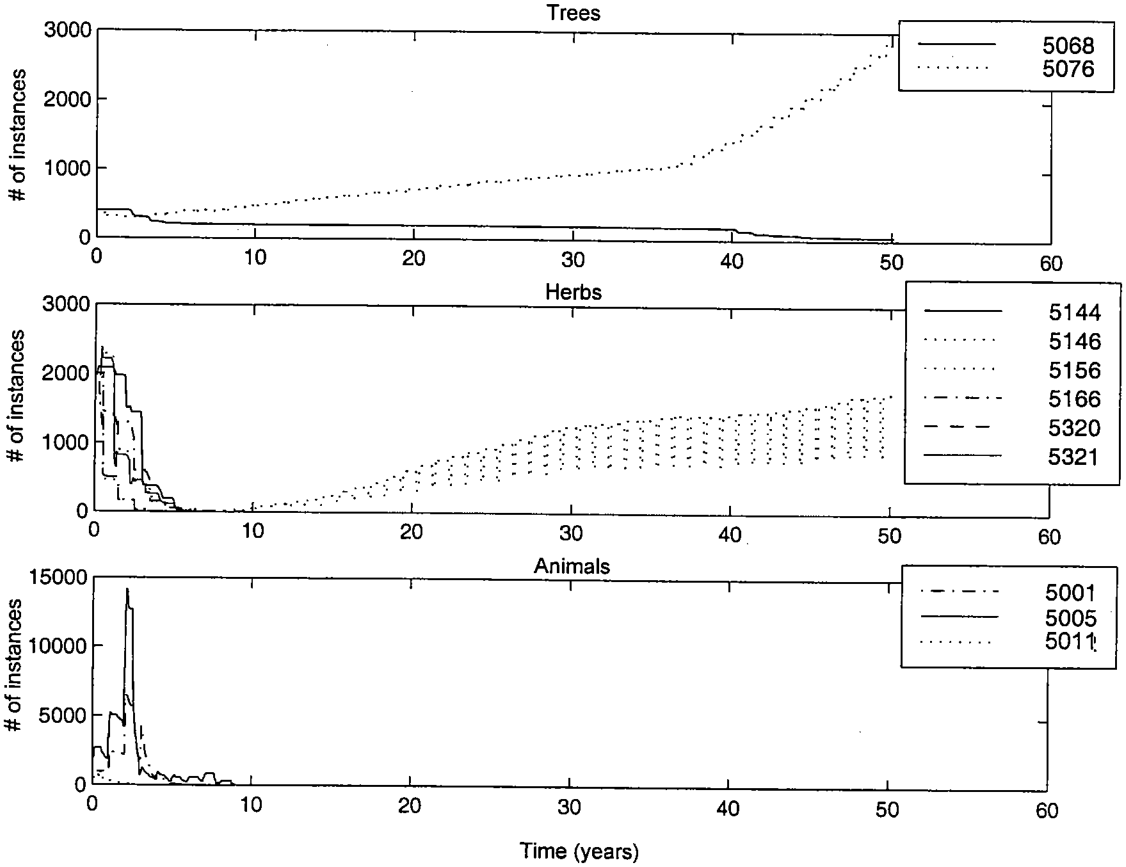


Figure 6.8: Population dynamics for AnimSim2.

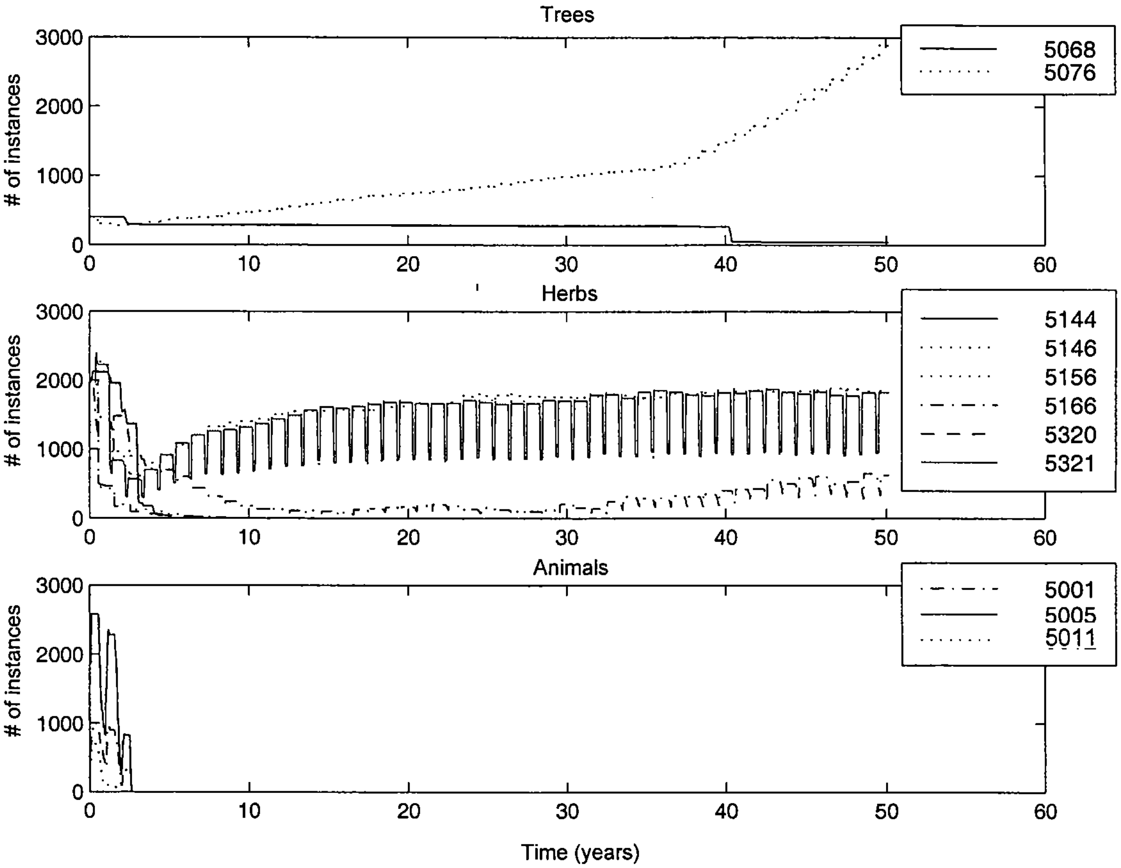


Figure 6.9: Population dynamics for AnimSim3.

compared to AnimSim2; in AnimSim3, two more herbaceous species survived and began to repopulate the terrain. The tree species do not appear to have been affected in any substantial way by the dynamics of the other species in the system.

AnimSim4 was run for testing purposes, to address a question: Why didn't the few remaining herbivores in AnimSim2 begin to repopulate the ecosystem after the grass species (#5146) had recovered? Based on a suspicion that perhaps the few, scattered individuals were not finding mates with which they could reproduce, a second simulation was run with the identical starting state as was used for AnimSim2, but with a slight modification to the FindMate method. As described above, in the original FindMate method, in order to reproduce, a female animal must find a mature male of the same species and move to its location on the terrain. This is effective when the population density is high, since the female does not usually have to travel very far. If, however, a female cannot reach the location of a potential mate during the current time increment, there is a fairly low probability that the female will continue to move towards the male in subsequent time increments since animals do not "remember" what they were doing in a previous time increment. Thus, AnimSim4 was run, based on a hypothesis that the female animals in the scattered herbivore populations were not reaching the males in order to reproduce. The FindMate routine was changed, therefore, so that as long as there was a mature male of the same species in the ecosystem, a female could become pregnant (i.e., the female and male do not have to be in the same location). The resultant population dynamics are shown in Figure 6.10.

In AnimSim4, two of the herbivorous mammalian species (#5005 and #5001) recover in numbers as the grass species regains dominance over the terrain. (The grass species that survives (#5321) is different from the one that survives in AnimSim2.) Thus, for these two herbivorous species, the limitation of the FindMate routine seems to have been hampering their ability to reproduce once food was plentiful again. Species #5011, however, does not recover; these

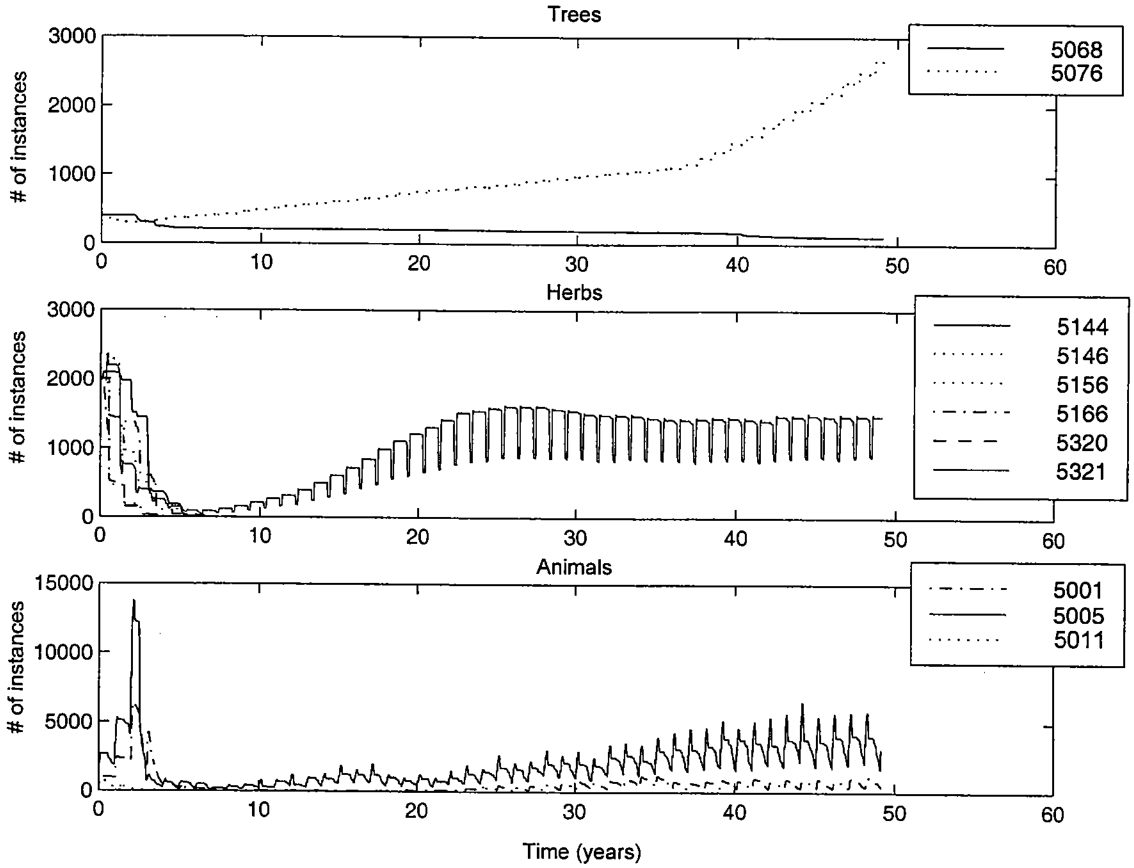


Figure 6.10: Population dynamics for AnimSim4.

animals eat mostly seeds, and do not have a very high food preference for grass. It is possible that they are not finding sufficient food to eat, since, although there are many trees in the ecosystem that produce seeds, the majority of these trees are probably seedlings. Species #5005 has a much larger population size than species #5001. This is due to their different attributes: Animals belonging to species #5001 are larger mammals (the average mass of a mature individual is 0.75kg) with a lower reproductive rate than the smaller animals (the average mass of a mature individual is 0.065kg) of species #5005.

### 6.7. Discussion & Conclusions

The ecosystem model can accommodate many more species than were included in AnimSim1, 2, 3 & 4. These simulations were initial experiments intended to test and verify the various animal routines and to assess the overall system response to the presence of animals. Overall, the results obtained show that the animal model performs reasonably well, and that both herbivorous and carnivorous species do persist in the system for a least a few years.

The markedly different population dynamics that arose in AnimSim4 as the result of a modification to one of the reproduction-related methods illustrates the strong impact that certain animal behaviours can have on the global level dynamics of an ecosystem. Although the assumption that was made in AnimSim4 with regards to reproduction was somewhat unrealistic, the simulation results did prove the validity of the hypothesis about the inability of the herbivore species to find mates when the population density is low. In the future, more realistic, reproduction methods could be tested. More significantly, however, the AnimSim4 experiment illustrates the flexibility of the model and the approach used to implement it. Once the model has been configured, it is easy to test the effects of changing one or more rules. This is a first version of the model; all of the methods can be later refined and modified as desired in order to achieve different results.

Of all the simulations performed with the model, both those reported here, as well as tens of others, none displayed three-trophic level population dynamics that was persistent over the long-term. In every case, the food web would collapse after a few years as the carnivores consumed all of the herbivores, and then starved to death. Even the two-trophic level systems were fragile: Often the primary producers would pay too great a toll as a result of predation, and many species would become either extinct or marginal. The lack of persistence of many species in these simple systems is not surprising: First, a transitory period is to be expected, in which the various species become established, and second, very few species were included in the ecosystems on which these simulations were based. In a natural food web, hundreds or thousands of interacting species have adapted together to fill ecological niches that support the survival of other species in all kinds of subtle ways, which are not accommodated by the model (Schulze and Mooney, 1994; Johnson et al., 1996). In future experiments with the model, therefore, more species should be included, and omnivores should be created that have lower food intake requirements than the carnivores that were tested. It is important to note, however, that even with the few species that were included, the ecosystem as a whole always persisted and remained viable over the long term.

The animal objects were modelled in a "mass-centric" manner, in which each internal process involved an exchange or transformation of mass. All of the mass flows in the animal body were based on a "pull" approach, whereby mass was transferred according to overriding teleonomic rules. An alternative would have been to use a "push" approach, in which mass was transferred according to rate equations that might be functions of concentration, etc. The pull approach was used since it seemed to be somewhat simpler to derive reasonable parameter values for this type of model (although many of the values had to be selected based on "best guesses", this was deemed simpler than guessing values for the rate of absorption of carbohydrate through the intestinal walls of various species, etc.). The drawback of this approach is, of course, that it is based on perceived need, rather than availability. It has, however, the advantage of ensuring that the animal's bodies have correct material compositions, but has the disadvantage that resources are not always fully used. For example, an animal is not able to assimilate digested protein into the body if it doesn't have a complimentary amount of digested fat, etc.

For these, and other reasons, the animal routines are simplistic, yet they do serve as an adequate first implementation. There are, of course, many ways that the representation of animal objects could be improved, and some of these will be addressed in future versions of the ecosystem model. Behavioural routines, for example, are fairly rudimentary, and not very species-specific. It would be interesting to add species specific behaviours such as food foraging and hoarding strategies, or territory marking, in order to investigate the effects that these have, if any, on both global-level system dynamics, and species persistence and resilience to extreme environmental disturbances. Despite the necessary simplifications, simulations with animal-based configurations exhibited remarkable examples of ecological self-organisation, including species persistence and the emergence of cyclical predator-prey dynamics. For these reasons, the object-based approach used here provides fascinating examples of the features of ecosystem comportment that are seen to be the aggregate result of many interacting components.

## 6.8. References

Beecham, J.A. and K.D. Farnsworth. 1998. Animal foraging from an individual perspective: an object orientated model. *Ecological Modelling* 113: 141-156.

Booth, G. 1997. Gecko: A continuous 2-D world for ecological modelling. *Artificial Life* 3(3): 147-163.

DeAngelis, D. and L. Gross (eds.). 1992. *Individual-based Models and Approaches in Ecology*. New York: Chapman and Hall.

Fahse, L., C. Wissel and V. Grimm. 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.

Friend, A.D, A.K. Stevens, R.G. Knox and M.G.R. Cannell. 1997. A process-based, terrestrial biosphere model of ecosystem dynamics (Hybrid v3.0). *Ecological Modelling* 95: 249-287.

Gustafson, E. and R. Gardner. 1996. The effect of landscape heterogeneity on the probability of patch colonization. *Ecology* 77: 94-107.

Henein, K., J. Wegner and G. Merriam. 1998. Population effects of landscape model manipulation on two behaviourally different woodland small mammals. *Oikos* 81: 168-186.

Hill, Richard W. 1976. *Comparative Physiology of Animals: An Environmental Approach*. New York: Harper & Row Publishers.

Hogeweg, P. and B. Hesper. 1990. Individual-oriented modelling in ecology. *Mathematical and Computational Modelling* 13(6): 83-90.

Holland, John H. 1975. *Adaptation in Natural and Artificial Systems. An introductory analysis with applications to biology, control, and artificial intelligence.*

Ann Arbor, MI: University of Michigan Press.

Hraber, P.T. and B.T. Milne. 1997. Community assembly in a model ecosystem. *Ecological Modelling* 103: 267-285.

Huston, M., D. DeAngelis and W. Post. 1988. New computer models unify ecological theory. *Bioscience* 38: 682-691.

Johnson, K., K. Vogt, H. Clark, O. Schmitz and D. Vogt. 1996. Biodiversity and the productivity and stability of ecosystems. *Trends in Ecology and Evolution* 11(9): 372-377.

Judson, O. 1994. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution* 9(1): 9-14.

Letcher, B. H., H. A. Priddy, J. R. Walters, and L. B. Crowder. 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.

Lett, C., C. Silber and N. Barret. 1999. Comparison of a cellular automata network and an individual-based model for the simulation of forest dynamics. *Ecological Modelling* 121: 277-293.

Miramontes, O., R. Solé and B. Goodwin. 1993. Collective behaviour of random-activated mobile cellular automata. *Physica D* 63: 145-160.

Parrott, L. and R. Kok. 2000a. Implementation of a generally configurable object-based ecosystem model. Chapter 3 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000b. Baseline performance of an ecosystem model and simulation for the EcoCyborg Project. Chapter 4 of Parrott, L. 2000 *Learning to*

*Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Parrott, L. and R. Kok. 2000c. A generic primary producer model for use in ecosystem simulations. Chapter 5 of Parrott, L. 2000 *Learning to Engineer Life: Development of a generally configurable model for the simulation of artificial ecosystems.* Ph.D. diss., Dept. of Agricultural and Biosystems Engineering, McGill University, Montreal.

Robbins, C. 1993. *Wildlife Feeding and Nutrition,* 2nd Ed. San Diego, CA: Academic Press.

Schulze, E.D. and H.A. Mooney, eds. 1994. *Biodiversity and Ecosystem Function.* Berlin: Springer-Verlag.

Shugart, H.H., T.M. Smith and W.M. Post. 1992. The potential for application of individual-based simulation models for assessing the effects of global change. *Annual Review of Ecological Systems* 23: 15-38.

Swan, Henry. 1974. *Thermoregulation and Bioenergetics.* New York: American Elsevier Pub. Co.

Taylor, C., D. Jefferson, S. Turner and S. Goldman. 1988. RAM: Artificial life for the exploration of complex biological systems. In *Artificial Life,* ed. C. Langton, 276-295. Reading, MA: Addison-Wesley.

## 6.9. Appendix I - Set of variables used to describe the attributes and current states of animal objects

See next page.

## 6.9: Appendix I: Set of attributes used to describe the state of an animal object

| Attribute name | Description | Units |
|---|---|---|
| ID.NUMBER | instance identifier | - |
| STATUS | alive or dead | - |
| WHAT.ACT.METHOD | act method identifier | - |
| WHAT.GROW.METHOD | grow method identifier | - |
| ID.LEVEL1 | ancestor class | - |
| ID.LEVEL2 | ancestor class | - |
| ID.LEVEL3 | ancestor class | - |
| ID.LEVEL4 | ancestor class | - |
| ID.LEVEL5 | ancestor class | - |
| ANIMAL.DEVSTAGE | development stage | - |
| AGE | age | s |
| GENDER | male or female | - |
| ACTIVITY | current activity (sleeping, moving, eating, etc.) | - |
| BIRTHDATE | "date" at which a new animal object is created | s |
| X.COORDINATE | x coordinate | - |
| Y.COORDINATE | y coordinate | - |
| X.COORD.HOUSE | x coordinate of "house" | - |
| Y.COORD.HOUSE | y coordinate of "house" | - |
| IS.LUMP | true or false | boolean |
| NUM.UNITS | number of units in a lumped animal | - |
| MASS | total mass | kg |
| UNIT.MASS | mass of one unit in a lumped animal | kg |
| BIRTHMASS | mass of animal at birth/creation | kg |
| MASS.NITROGEN | mass of nitrogen in body mass | kg |
| MASS.CARBON | mass of carbon in body mass | kg |
| MASS.OXYGEN | mass of oxygen in body mass | kg |
| MASS.HYDROGEN | mass of hydrogen in body mass | kg |
| MASS.CARBO | mass of carbohydrate in body mass | kg |
| MASS.FAT | mass of fat in body mass | kg |

| Attribute name | Description | Units |
|---|---|---|
| MASS.PROTEIN | mass of protein in body mass | kg |
| MASS.WATER | mass of water in body mass | kg |
| LEAN.MASS | mass of lean mass | kg |
| FATTY.MASS | mass of fatty mass | kg |
| LM.PROT | mass of protein in lean mass | kg |
| LM.WATER | mass of water in lean mass | kg |
| FAT.WATER | mass of water in fatty mass | kg |
| FAT.FAT | mass of fat in fatty mass | kg |
| MASS.STOM.FOOD | total mass of food currently in stomach | kg |
| STOMACH.FAT | mass of fat in food in stomach | kg |
| STOMACH.CARBO | mass of carbohydrate in food in stomach | kg |
| STOMACH.PROTEIN | mass of protein in food in stomach | kg |
| STOMACH.WATER | mass of water in food in stomach | kg |
| DIGEST.CARBO | mass of digested carbohydrate | kg |
| DIGEST.FAT | mass of digested fat | kg |
| DIGEST.PROT | mass of digested protein | kg |
| DIGEST.WATER | mass of digested water | kg |
| LIQUID.WASTE | liquid mass (water) to be excreted | kg |
| DRY.WASTE | total dry mass to be excreted (carbo, fat & protein) | kg |
| WASTE.FAT | mass of fat to be excreted | kg |
| WASTE.WATER | mass of water to be excreted | kg |
| WASTE.CARBO | mass of carbohydrate to be excreted | kg |
| WASTE.PROTEIN | mass of protein to be excreted | kg |
| BABY.MASS | mass in "womb" of female animal | kg |
| OPT.FATTY.MASS | optimum fraction of body mass that is fatty mass | - |
| OPT.LEAN.MASS | optimum fraction of body mass that is lean mass | - |
| MAX.FATTY.MASS | maximum fraction of body mass that can be fatty mass | - |
| OPT.LM.WATER | optimum fraction of lean mass that is water | - |
| OPT.LM.PROTEIN | optimum fraction of lean mass that is protein | - |
| OPT.FAT.WATER | optimum fraction of fatty mass that is water | - |
| OPT.FAT.FAT | optimum fraction of fatty mass that is fat | - |

| Attribute name | Description | Units |
|---|---|---|
| LM.STOMACH.CAP | maximum capacity of stomach, as a fraction of lean mass | - |
| BMETABOLIC.RATE | basal metabolic rate | kcal·Δt-1 |
| A | metabolic rate constant | - |
| B | metabolic rate constant | - |
| RESP.RATE | current respiration rate | kgO2·Δt-1 |
| RESP.CARBO | mass of carbohydrate to be respired in current time step | kg |
| MAX.RESP.RATE | maximum respiration rate in one time step | kgO2·Δt-1 |
| DIGESTION.RATE | digestion rate of food in stomach | kg·s-1 |
| FAT.ACCUM.RATE | rate at which fatty mass is accumulated | kg·s-1 |
| BABY.GROW.RATE | rate at which mass is transferred to body to support gestation | kg·s-1 |
| POTLEANMASS | potential lean mass of animal at current age | kg |
| MASS.GROW.RATE | potential growth rate of juvenile animal | kg·s-1 |
| K1 | growth rate constant | - |
| K2 | growth rate constant | - |
| K3 | growth rate constant | - |
| MAXIMUM.SPEED | max. travel speed of animal | m·s-1 |
| DIRECTION | direction in which animal is moving | - |
| FOOD.SEARCH.RAD | maximum radius of food search area | m |
| ID.OF.FOOD | identifier of target object to be eaten | - |
| BUILDS.HOUSE | true if animal has a permanent "house" | Boolean |
| HAS.HOUSE | true if animal has a house | Boolean |
| IN.HOUSE | true if animal is inside its "house" | Boolean |
| CROWD.TOLERANCE | max. # of houses for the same species in a grid cell | - |
| MATURITY.AGE.M. | age of maturity for a male | s |
| MATURITY.AGE.F. | age of maturity for a female | s |
| MAXIMUM.AGE | lifespan | s |
| MATURE | true if animal's age is > age of maturity | Boolean |
| WEANING.AGE | age at which young leave their mother | s |
| REPRODUC.LIFE | maximum age at which a female can reproduce | s |
| START.REPRODUCT | time of year that female's reproductive period begins | s |
| REPRODUC.CYCLES | number of times a female can reproduce in a year | - |

| Attribute name | Description | Units |
|---|---|---|
| REPRODUC.TIME | length of each reproductive period | s |
| PREGNANT | true if animal is gestating | Boolean |
| CONCEPTION.DATE | date that a female becomes pregnant | s |
| GESTATION.TIME | length of gestation period | s |
| LITTER.SIZE | number of babies produced at end of a gestation period | - |
| HAS.YOUNG | true if female has recently given birth | Boolean |
| TIED.TO.YOUNG | if true, an animal must stay near her young | Boolean |
| ID.OF.MOTHER | identifier of animal's "mother" | - |
| ID.OF.MATE | identifier of a female's mate | - |
| HAS.MATE | true if a female has found a mate | Boolean |
| HUNGER | degree of hunger (range 0-1) | - |
| THIRST | degree of thirst (range 0-1) | - |
| SATIATION | fullness of stomach (range 0-1) | - |

## 6.10. Appendix II - Comparison of AnimSim1 with the baseline simulations

The overall comportment of the model when configured with animal species in the biological component realm was compared with the preliminary baseline simulations reported in Parrott and Kok (2000b, Figures 9-12). In Figure 6.11 and Figure 6.12, the histories of the masses of the ecosystem sections, and of the masses of the compounds in the ecosystem for AnimSim1 are shown. In these, for all mass forms affected by the presence of the biological components, long cycles can be observed, which correspond to the cyclic population dynamics that emerged in this simulation (Figure 6.7). In comparison to the Baseline2 and Baseline3 simulations, there are several notable observations: First, the total biomass in AnimSim1 (Figure 6.11) stabilizes at an average of $2\times10^6$ kg, which is considerably lower than that observed for the strictly plant based ecosystems. This is likely due to the sparsity of species that remain in AnimSim1, and due to the continual grazing by herbivores which limits the long-term accumulation of plant biomass in this system. Second, the presence of herbivores seems to dampen the amplitude of the yearly biomass fluctuations that occurred in the Baseline2 and Baseline3 experiments. Third, some differences in the histories of the compounds can be noted for the three simulations. For example, in AnimSim1, there is a persistent amount of fat resident in the system, due to the presence of animal objects. In terms of overall trends, the system in AnimSim1 is qualitatively similar to that of Baseline2, since both of these show fairly stable trends in the distribution of compounds over time. In the high plant biomass system of Baseline3, however, the carbon dioxide levels can be seen to decrease steadily, in tandem with corresponding increases in oxygen levels, due to plant photosynthesis. In contrast, the animals in AnimSim1 appear to have a moderating influence on the effects of photosynthesis, and the levels of carbon dioxide and oxygen remain relatively steady in this system.

These results fairly clearly illustrate the importance of the presence of animal type organisms in an ecosystem, particularly a closed system. It is remarkable that, even with the use of very simplistic rules describing animal energetics and

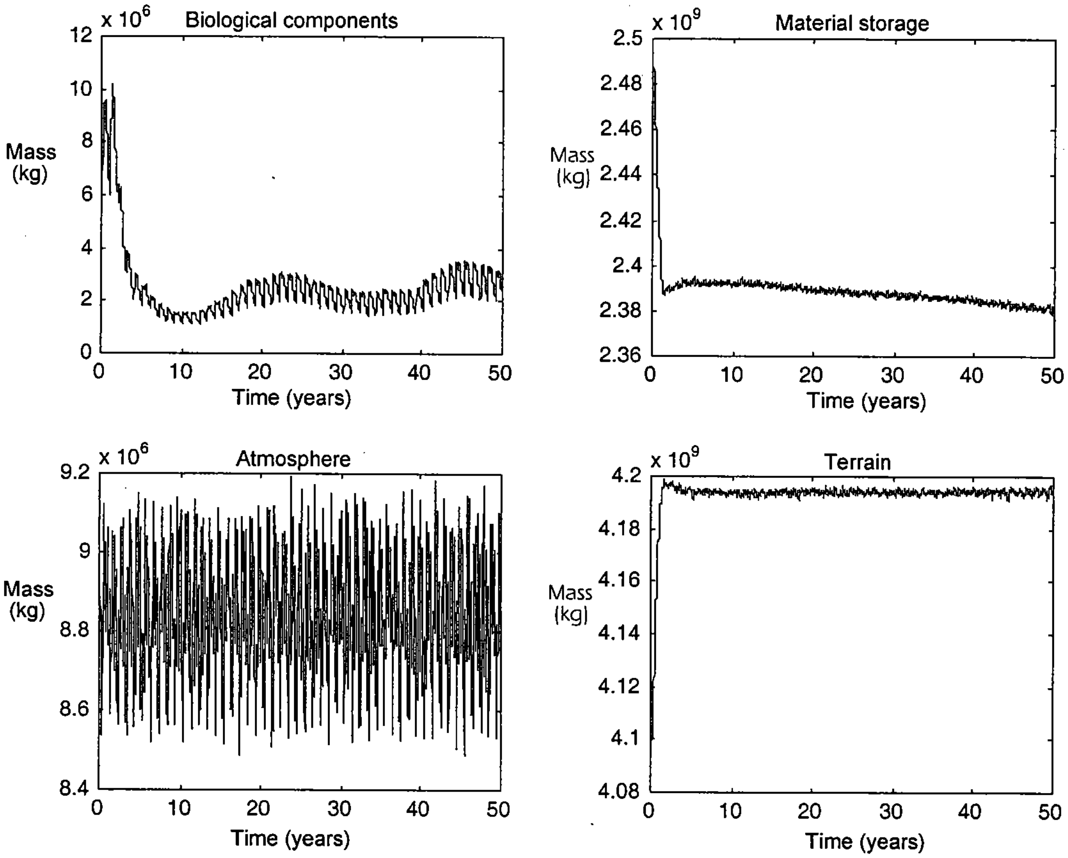


Figure 6.11: Total masses of the ecosystem sections for AnimSim1.

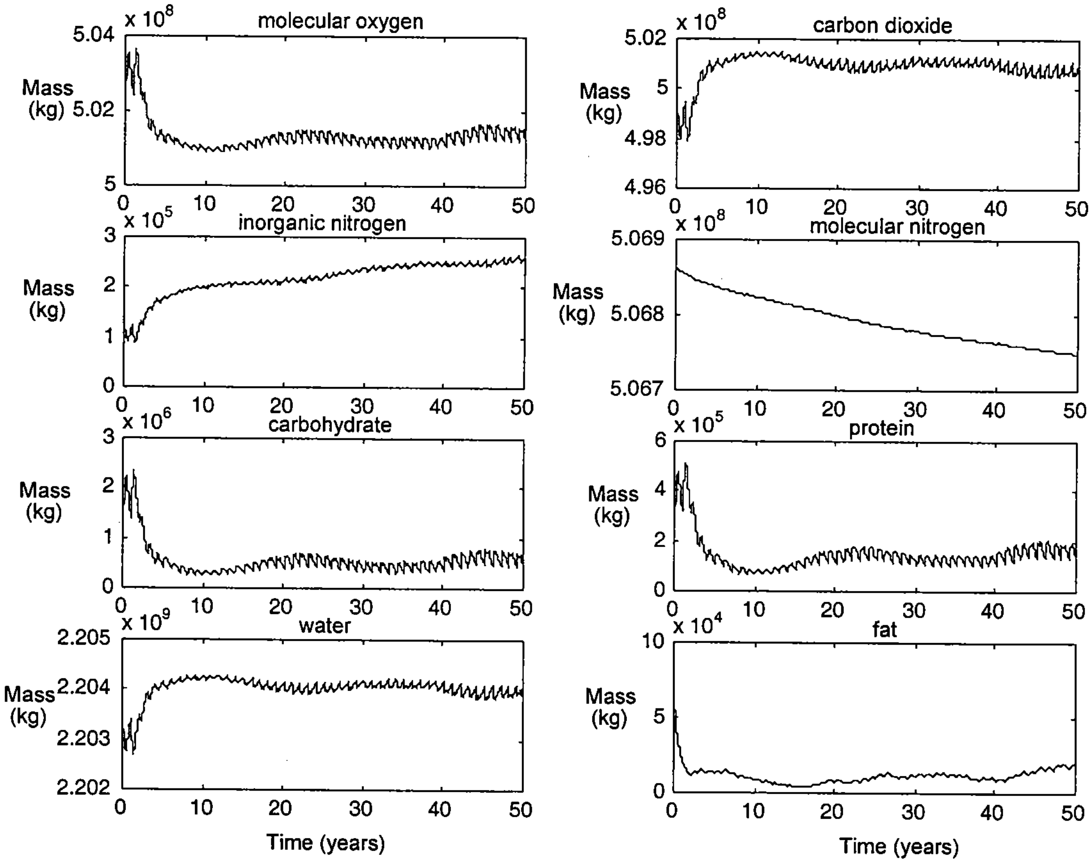


Figure 6.12: Total masses of the compounds in the ecosystem for AnimSim1.

behaviour, an overall influence on the global system state due to the presence of animal objects can still be observed. Although most of the biospheric, global climate models do not include considerations of material flow due to animal respiration, the preliminary results obtained here would suggest that animals do have an impact on the global system state and that their presence must be considered, especially for models of small-scale systems.

## Chapter 7. General Discussion and Conclusions

The overall objective of this project was to create a model with which the behaviour of ecosystems could be simulated, and specifically, to reproduce some of the complex features that are typical of natural living systems, including autopoiesis and self-organisation. Accordingly, a model has been described that was developed to meet this objective, and sample simulations have been presented. In this chapter, some of the points that have been raised in previous discussions will be summarized and reiterated in the overall context of the project.

As discussed in Chapter 1, one of the most salient characteristics of a complex system is the presence of dissimilar spatial and temporal features at different scales. This makes a complex system difficult to describe since, for each observable regularity or feature at a particular scale, a different descriptive model can be written. Thus, one of the challenges in modelling and simulating such systems is to find a means by which as many of these features can be represented as possible. The object-based modelling approach that was presented in Chapter 2, and used for the development of this ecosystem model, is one approach that has been presented as a way to depict many of the features of a complex system over a range of scales.

In the model developed for this project, an ecosystem was represented at a variety of different levels of resolution: the level of the organism in the case of some species, and of the small lump of organisms for others; the level of the grid cell in the case of the terrain, etc. In addition, for each component that was represented as an object, some higher resolution information regarding the object's composition was retained (e.g., masses of the different parts of a plant, animal, or grid cell, and the corresponding compositions of these in terms of the various mass forms that are tracked in the system, etc.). Through this representation of the states of lower level components, and the subsequent implementation of these in simulation, many interesting features emerged at all scales, particularly those corresponding to the population and ecosystem levels.

The extent to which these features are indicative of complex, life-like behaviour is discussed in the following sections.

## 7.1. Evidence of complex features in simulations based on the ecosystem model

### *7.1.1. Autopoiesis: Persistence of species, regeneration of the biological community after a trauma*

The concept of autopoiesis, or self-regeneration, was introduced in Chapter 1, and was discussed with reference to the ability of biological systems to regenerate damaged components, thereby maintaining global level structures. Ecosystems, as well, are autopoietic systems: In an established ecosystem, the structure of the biological community and the distribution of biomass amongst the components of the system, is maintained in the presence of continuous environmental "noise". An ecosystem that undergoes a severe trauma, such as a forest fire or the loss of a keystone species, for example, will regenerate, resulting in a new, meta-stable state (Green, 1994). Similarly, in the ecosystem configurations that have been implemented in simulation and discussed here, dynamics were observed at both the population and system level that are autopoietic in nature. These are elucidated in the following paragraphs, with reference to the population dynamics of several species in different ecosystem configurations.

Two simulations (AnimSim2 and AnimSim4, see Chapter 6) have been selected as comparative examples, in which both the persistence of species, and the regeneration of the biological community after a trauma, can be observed. These two phenomena in an ecosystem may be interpreted as examples of autopoiesis at the population and system levels, respectively. The population dynamics in time for the two simulations were presented previously in Figures 6.8 and 6.10. In Figure 7.1(a-d) and Figure 7.2(a-d), the equivalent dynamics are illustrated in both time and space for selected species in the two systems (only those species that persisted in relatively significant numbers throughout most of the simulation are shown). In these figures, the relative abundance (in terms of

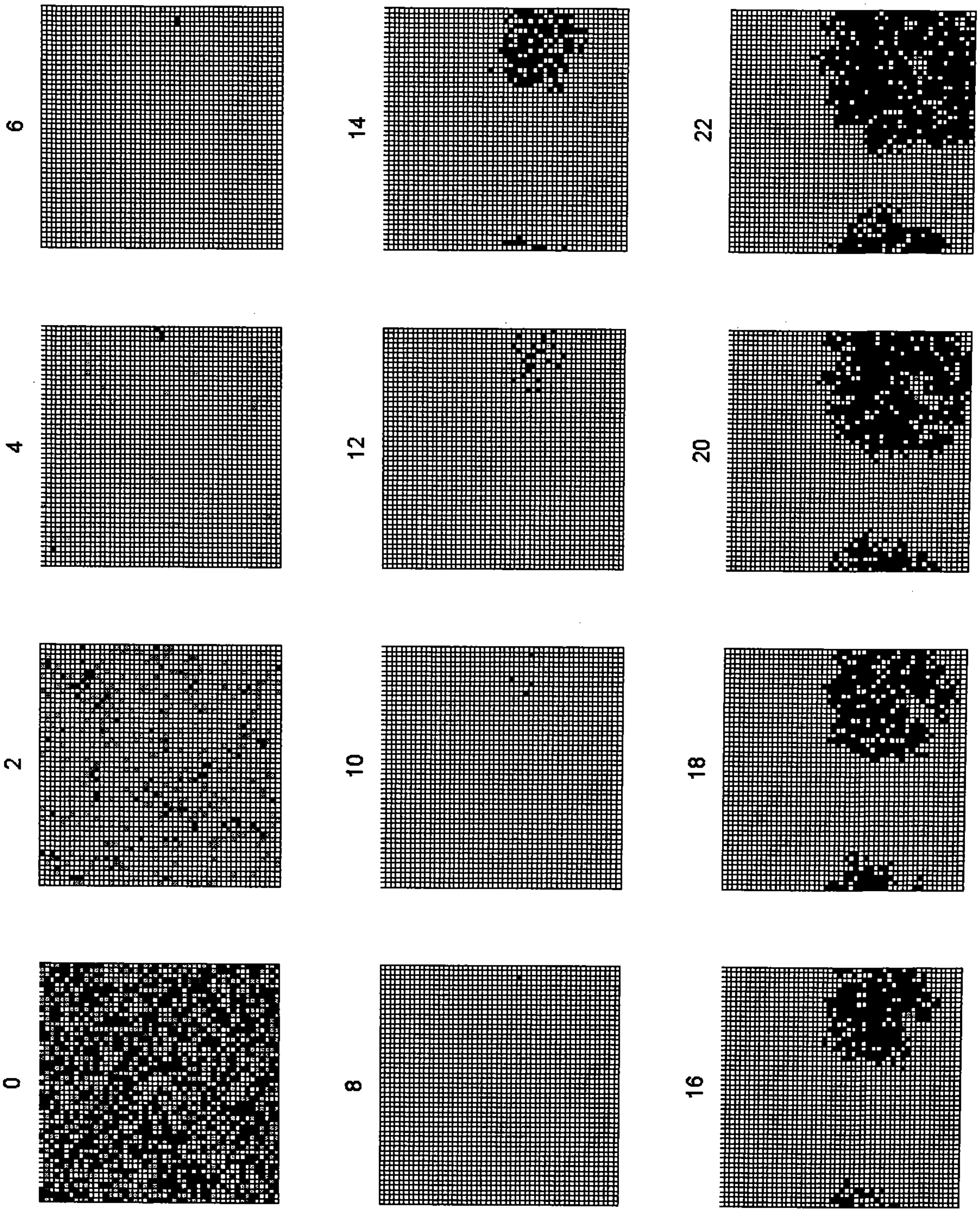


Figure 7.1(a): Population dynamics in space and time for perennial grass species #5146 in AnimSim2.

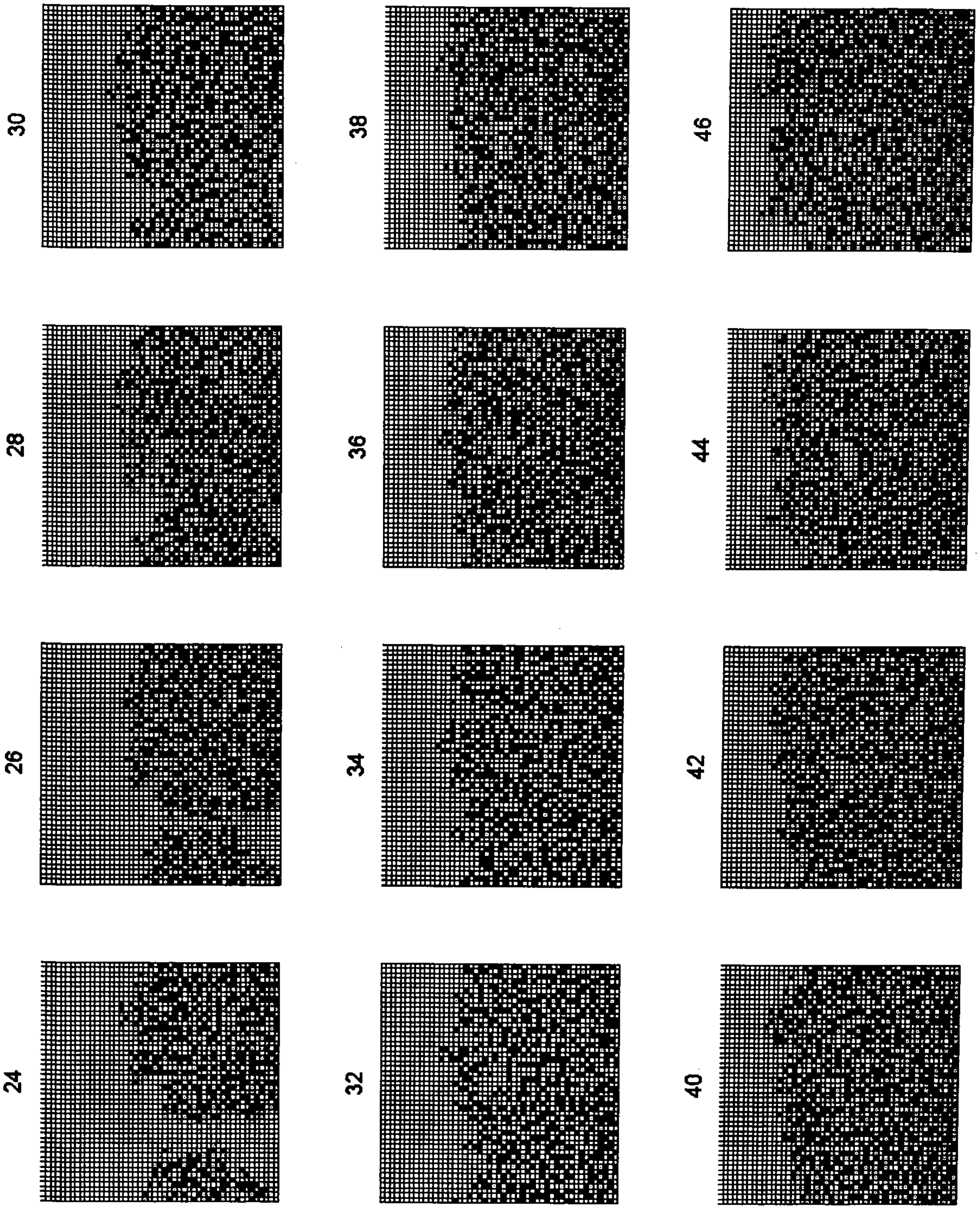


Figure 7.1(a) cont.

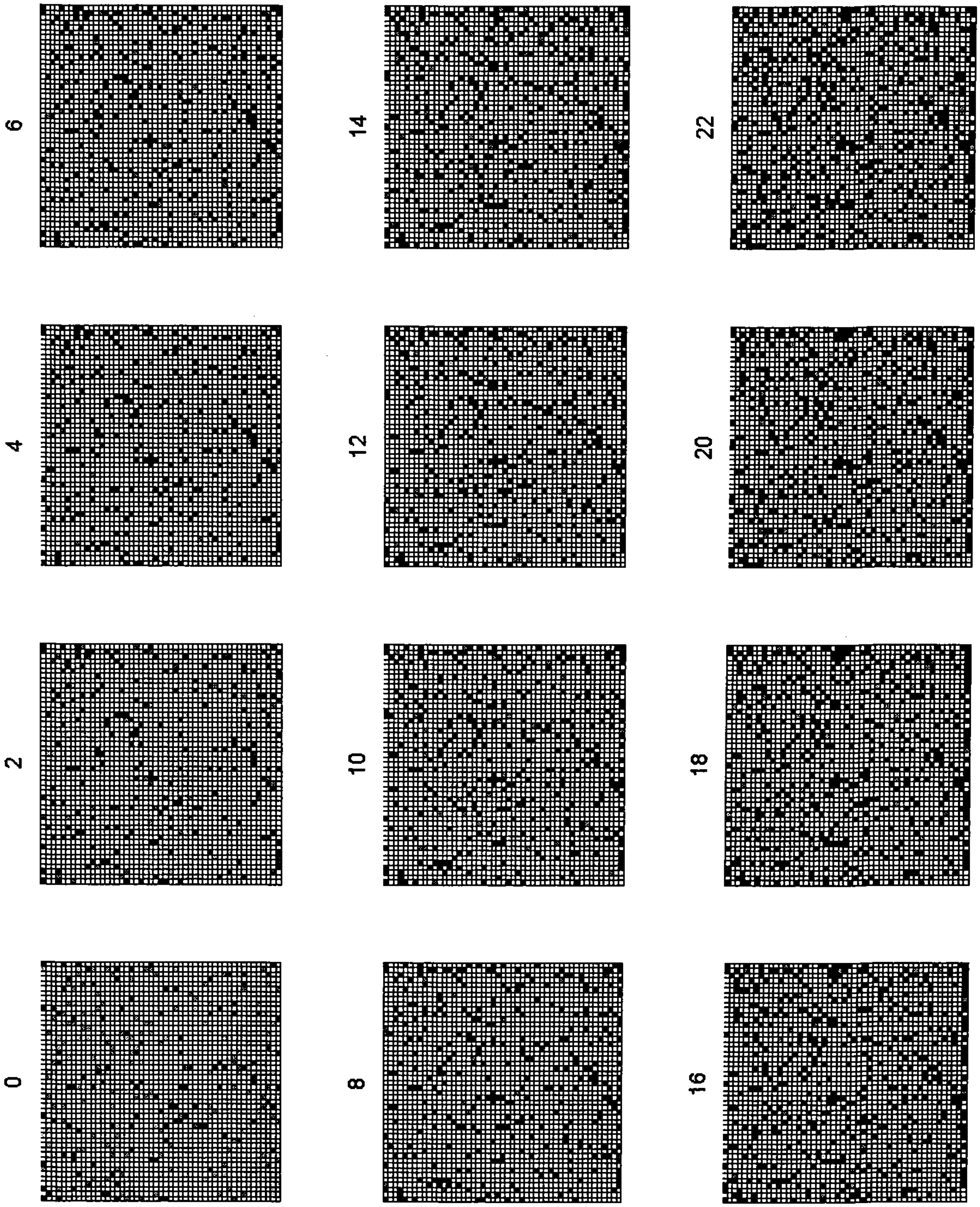


Figure 7.1(b): Population dynamics in space and time for deciduous tree species #5076 in AnimSim2.

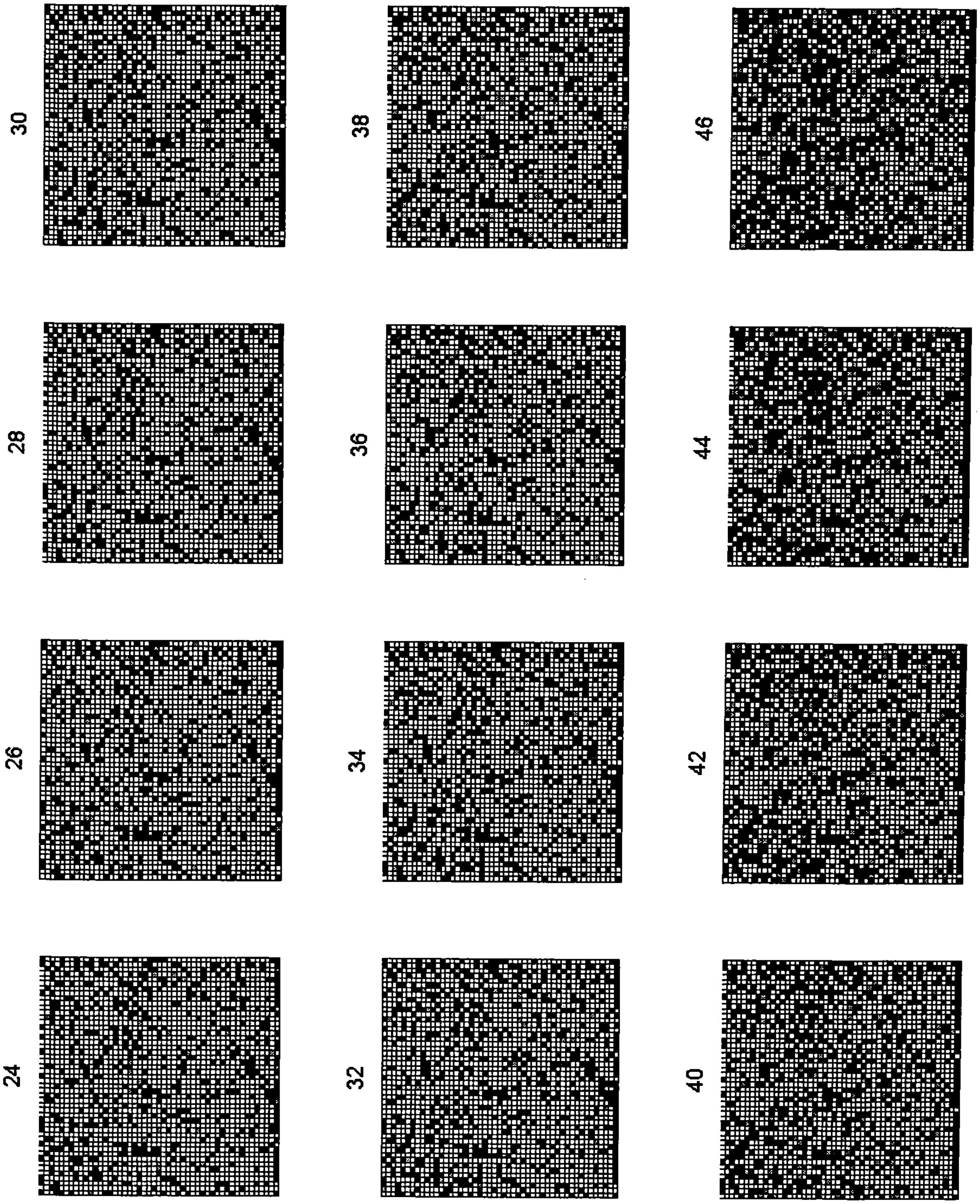


Figure 7.1(b) cont.

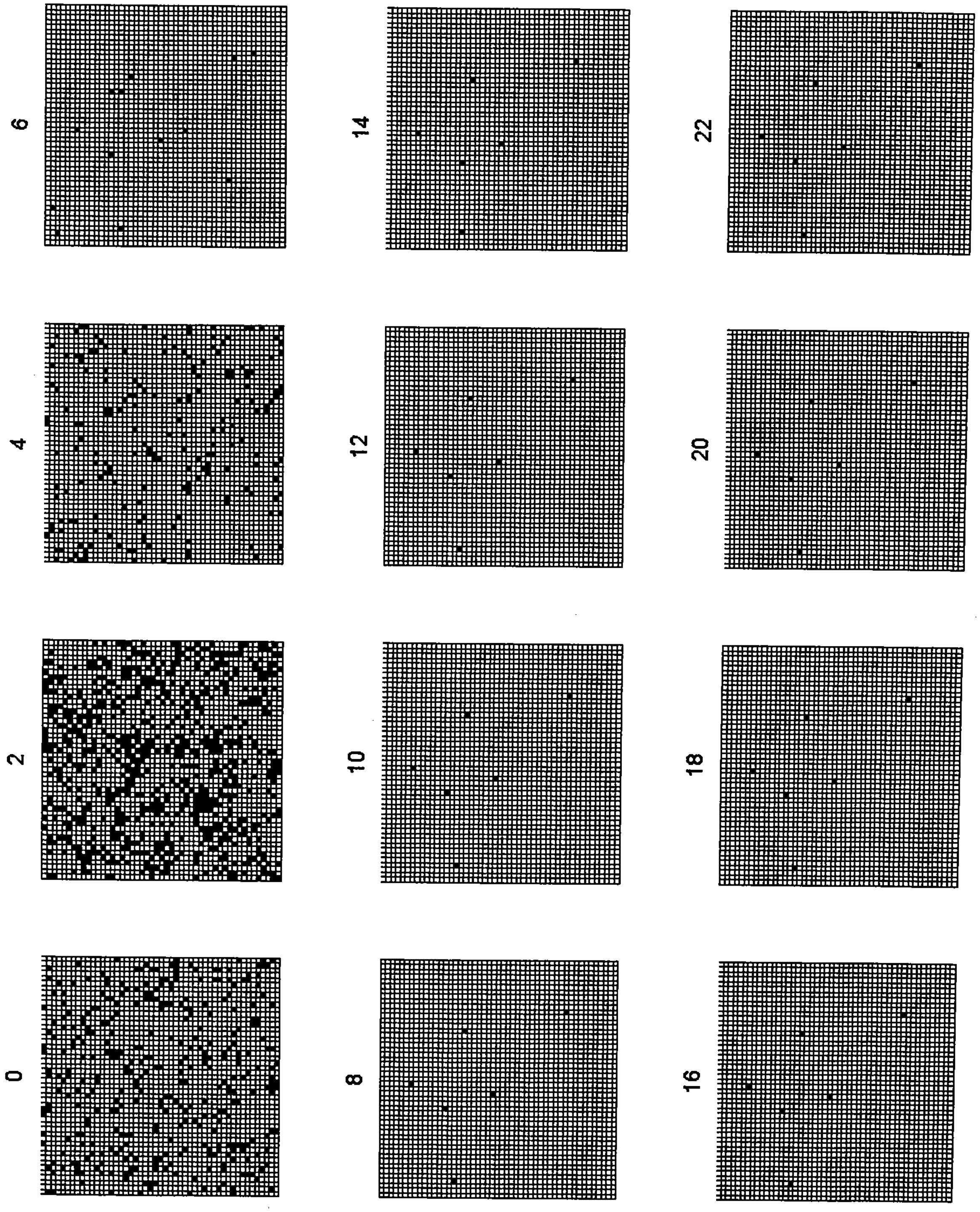


Figure 7.1(c): Population dynamics in space and time for herbivorous species #5001 in AnimSim2.

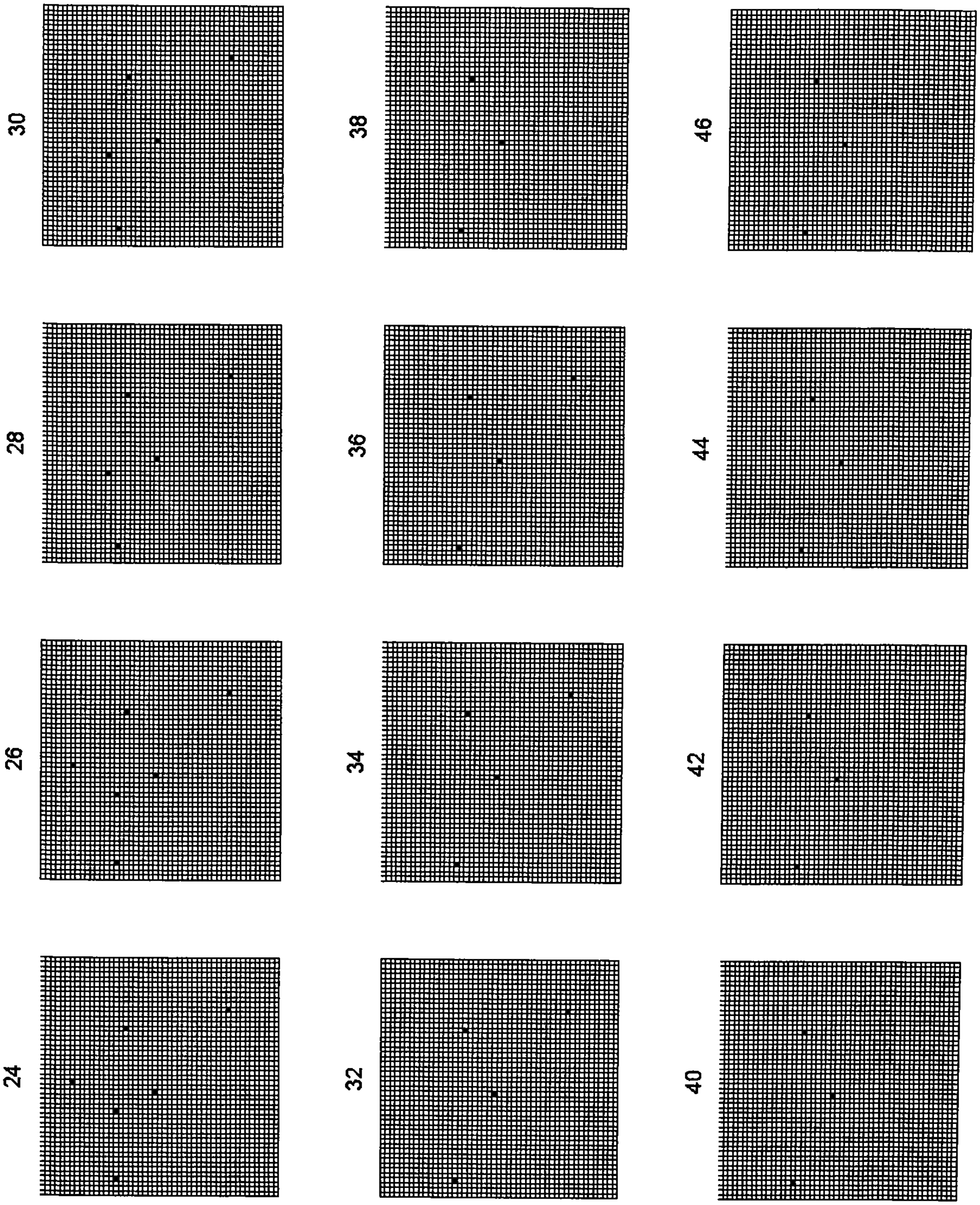


Figure 7.1(c) cont.

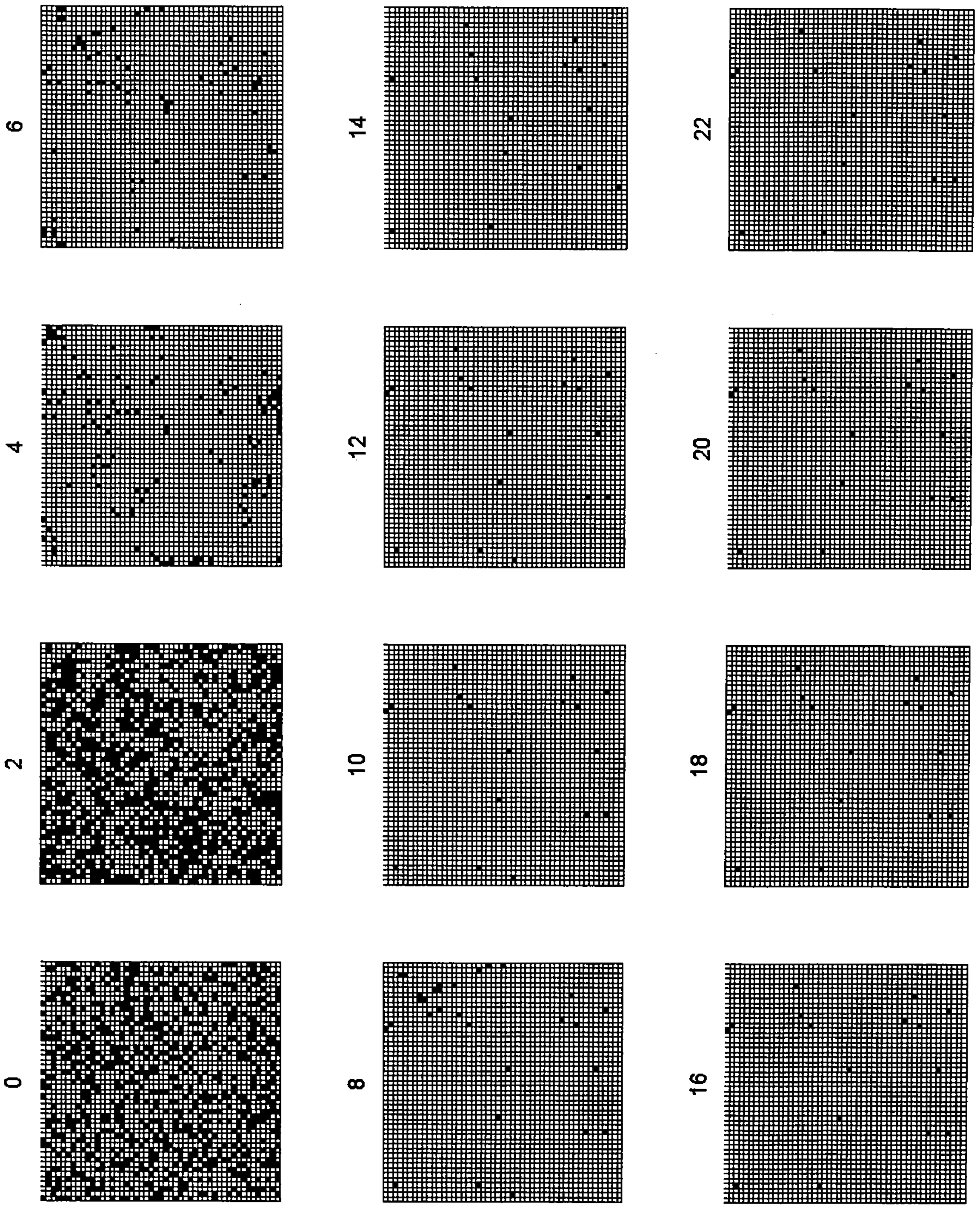


Figure 7.1(d): Population dynamics in space and time for herbivorous species #5006 in AnimSim2.

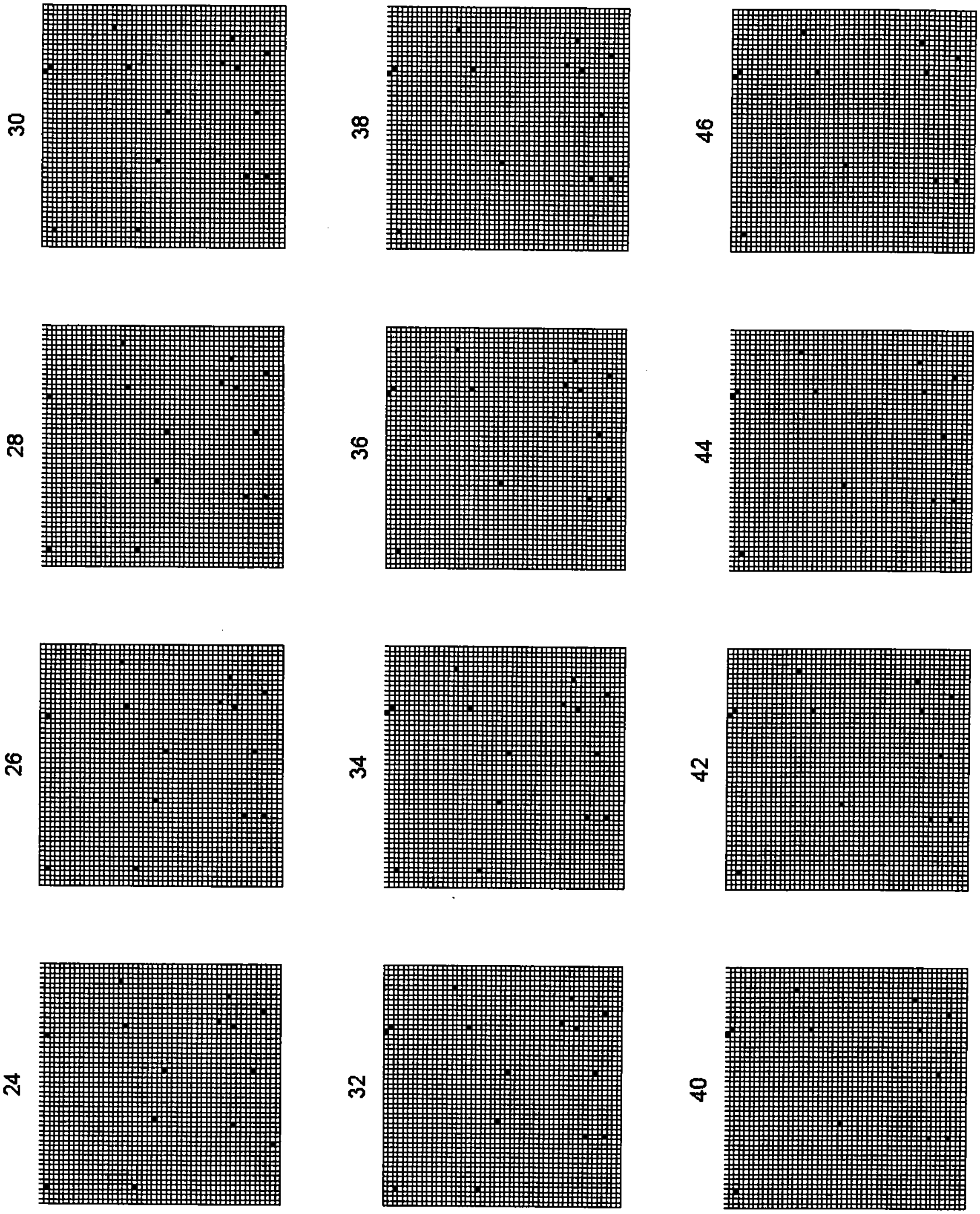


Figure 7.1(d) cont.

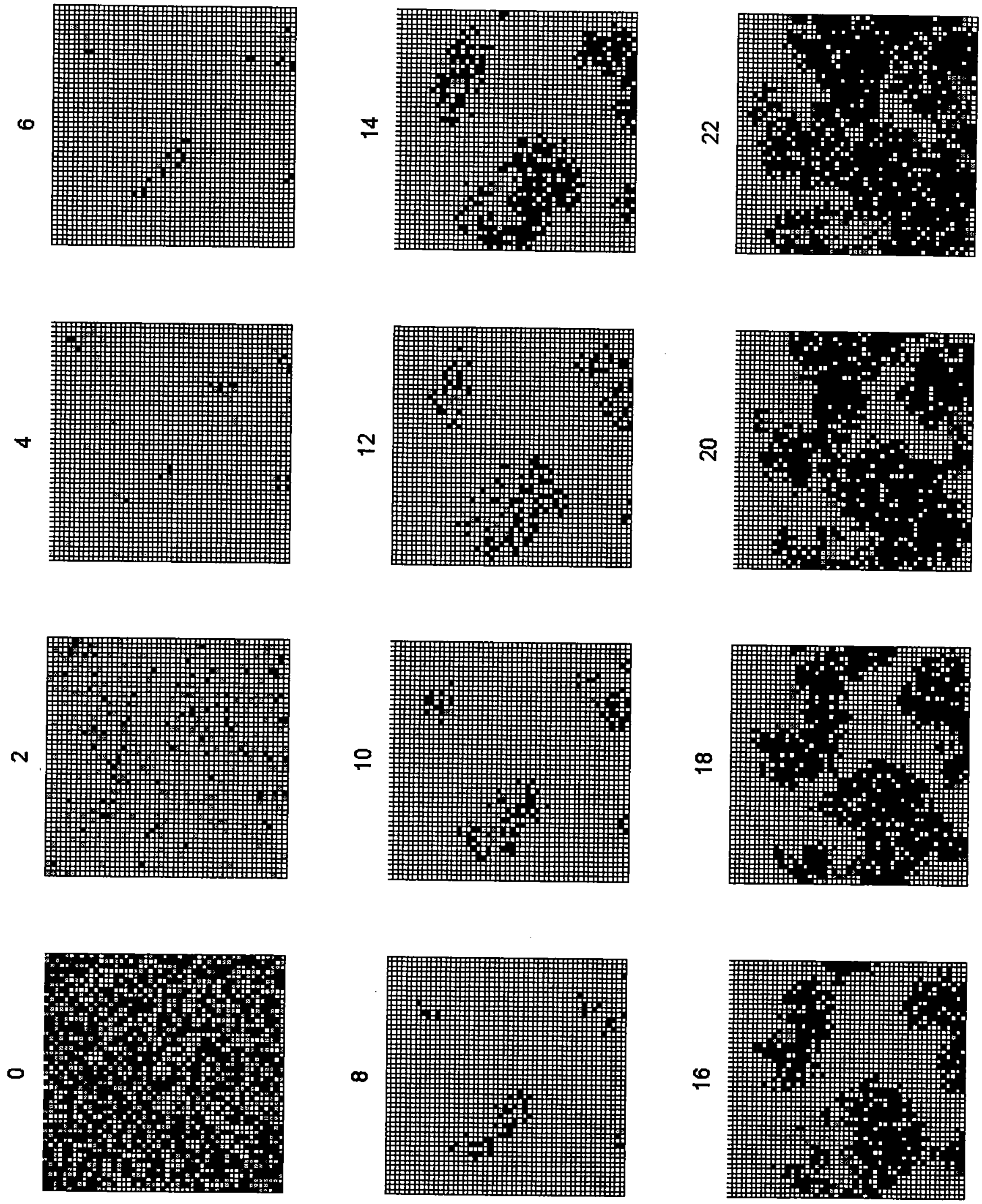


Figure 7.2(a): Population dynamics in space and time for annual grass species #5321 in AnimSim4.

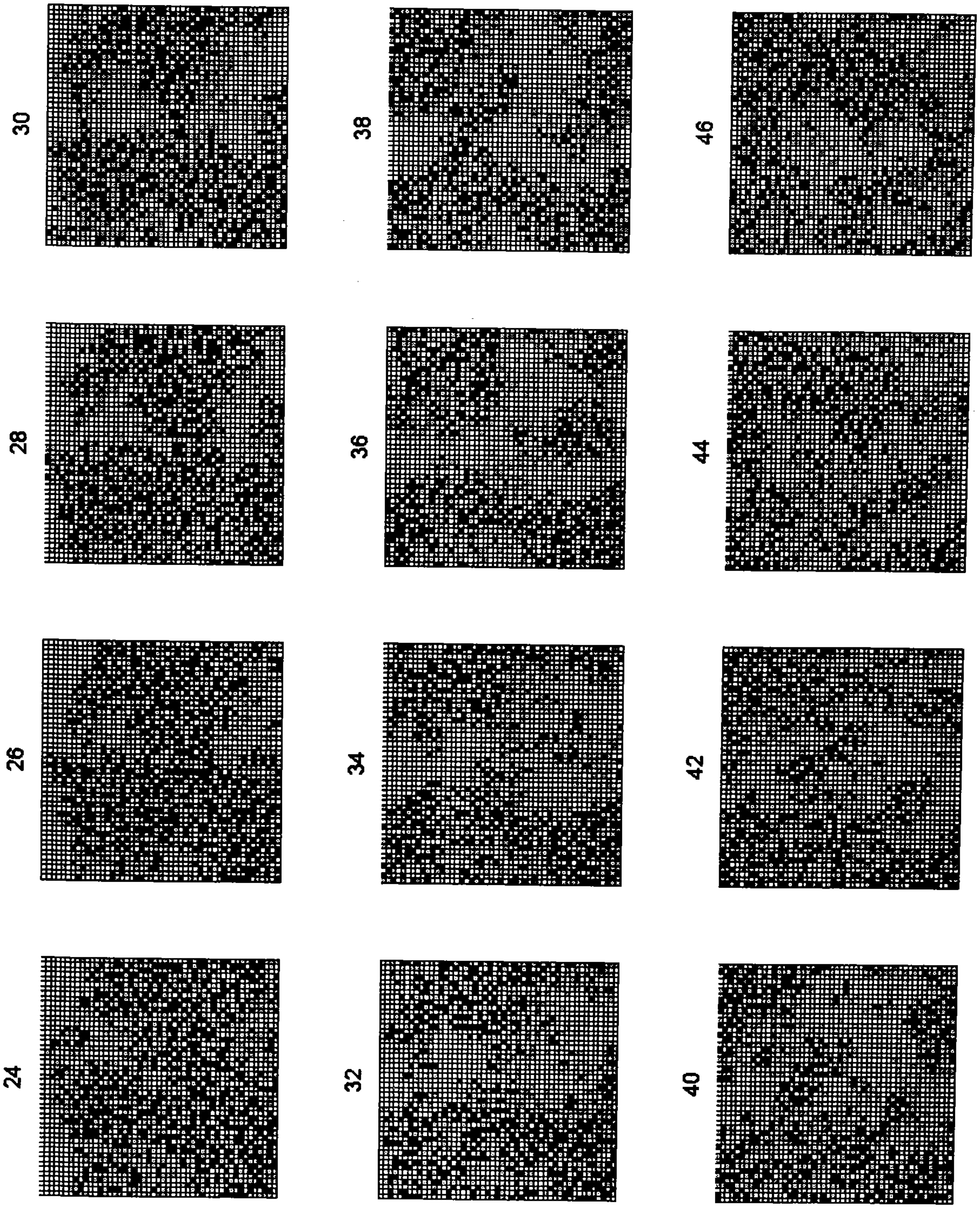


Figure 7.2(a) cont.

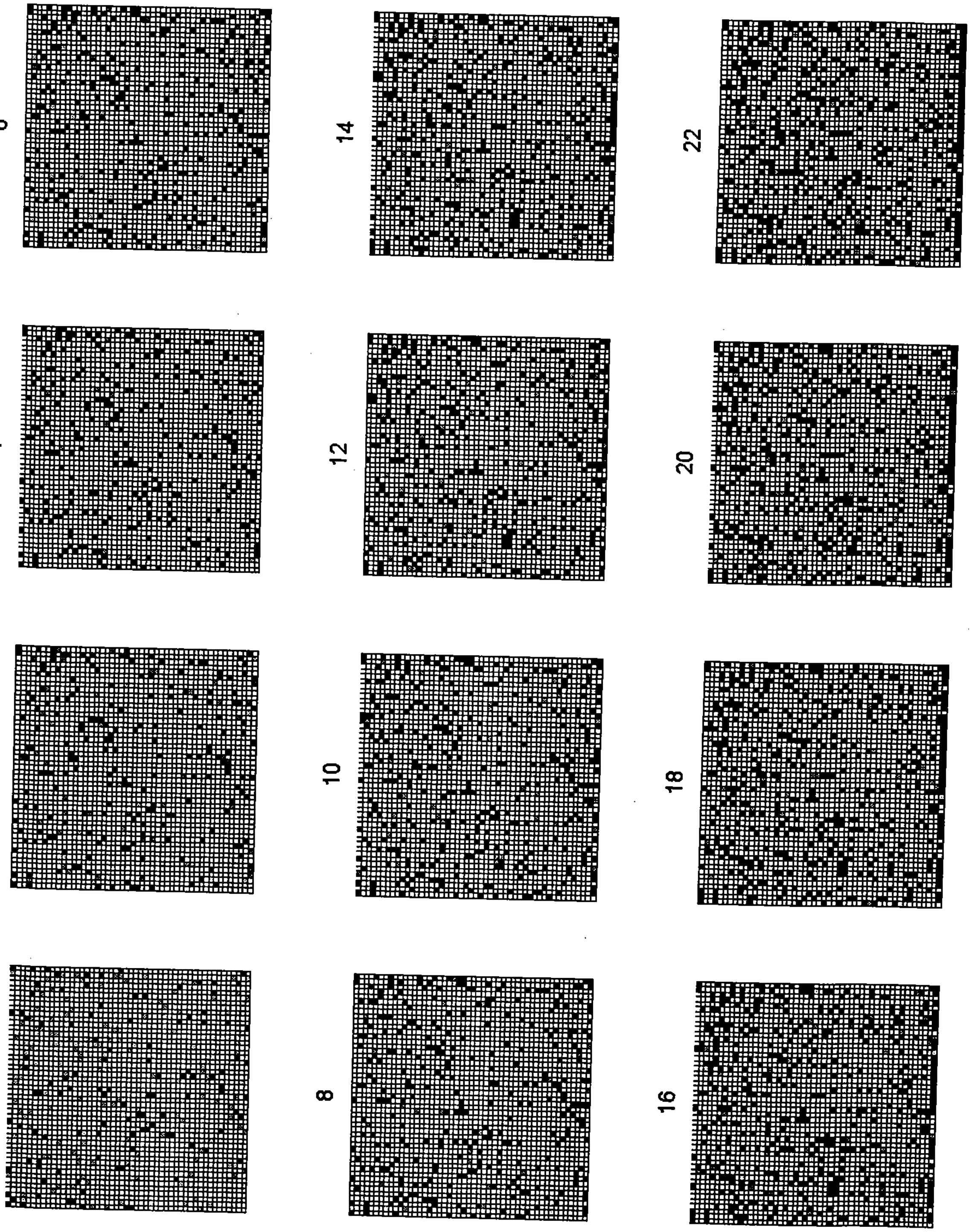


Figure 7.2(b): Population dynamics in space and time for deciduous tree species #5076 in AnimSim4.

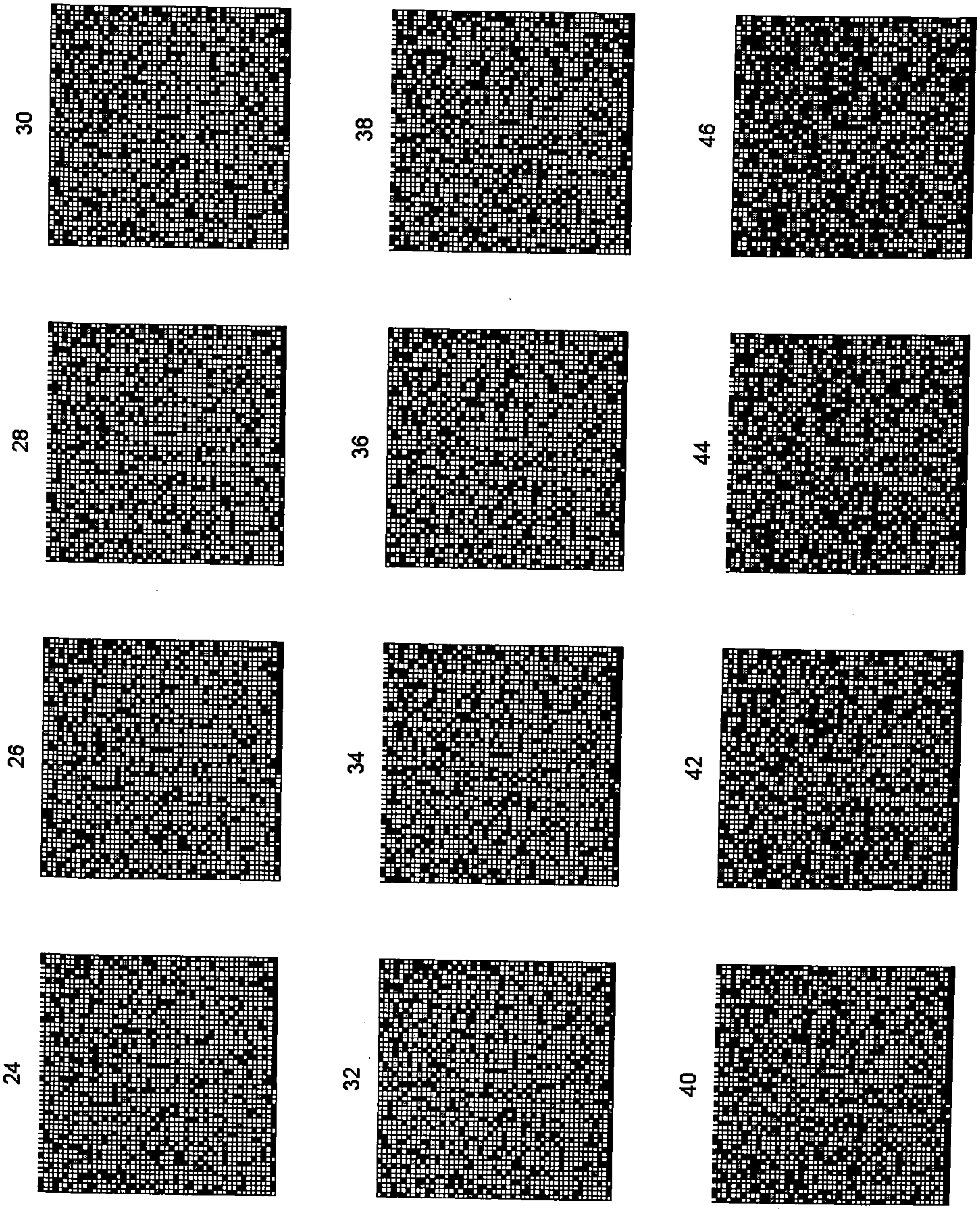


Figure 7.2(b) cont.

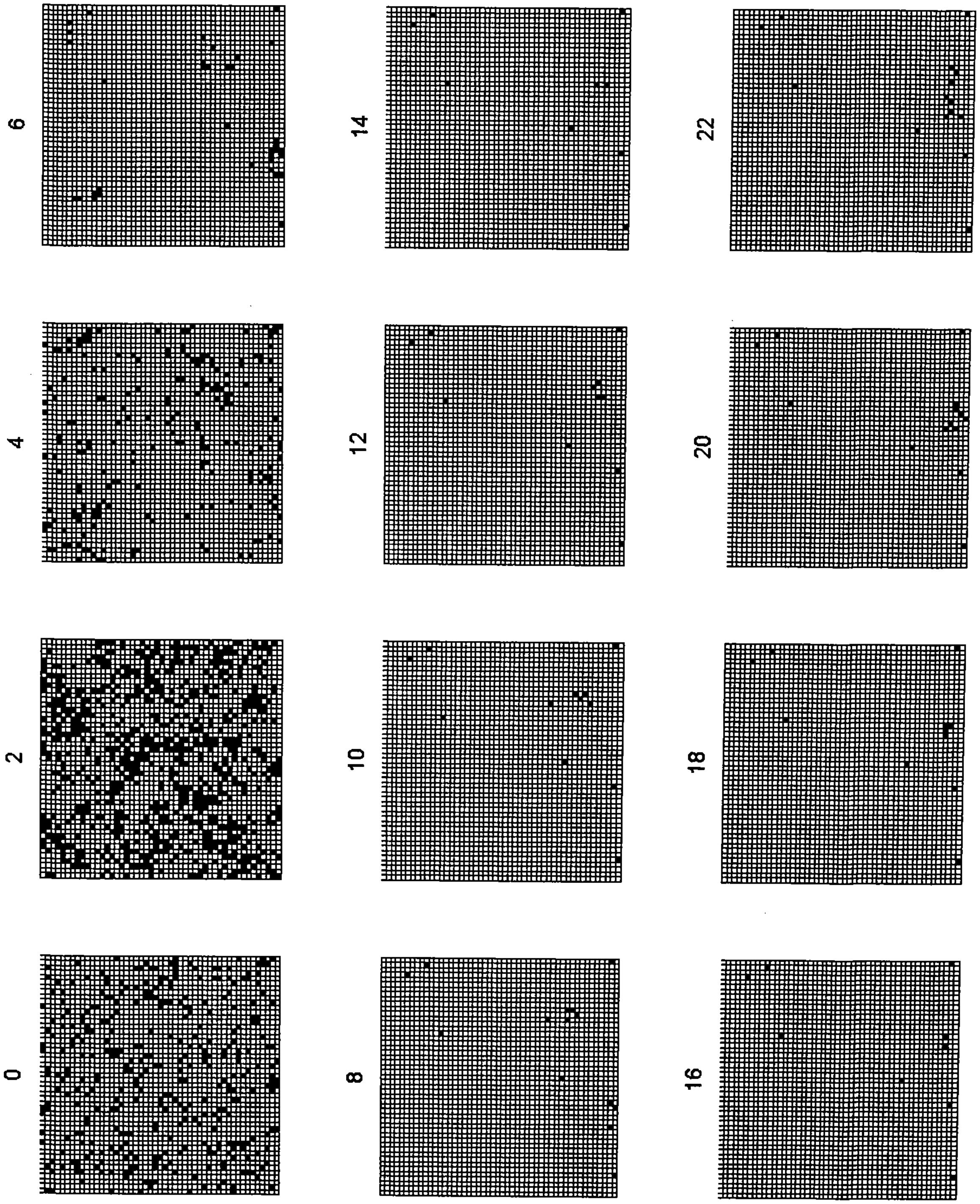


Figure 7.2(c): Population dynamics in space and time for herbivorous species #5001 in AnimSim4.

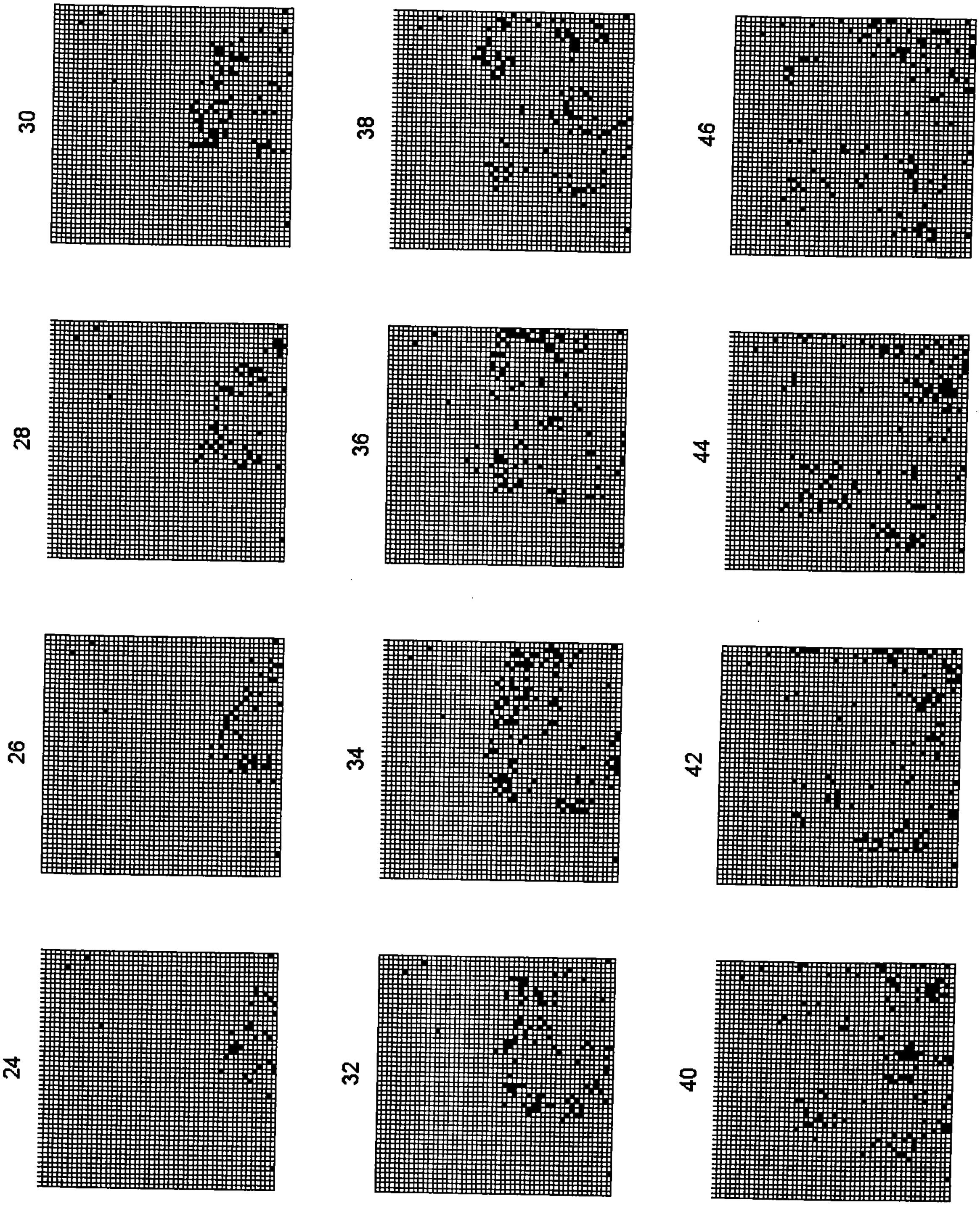


Figure 7.2(c) cont.

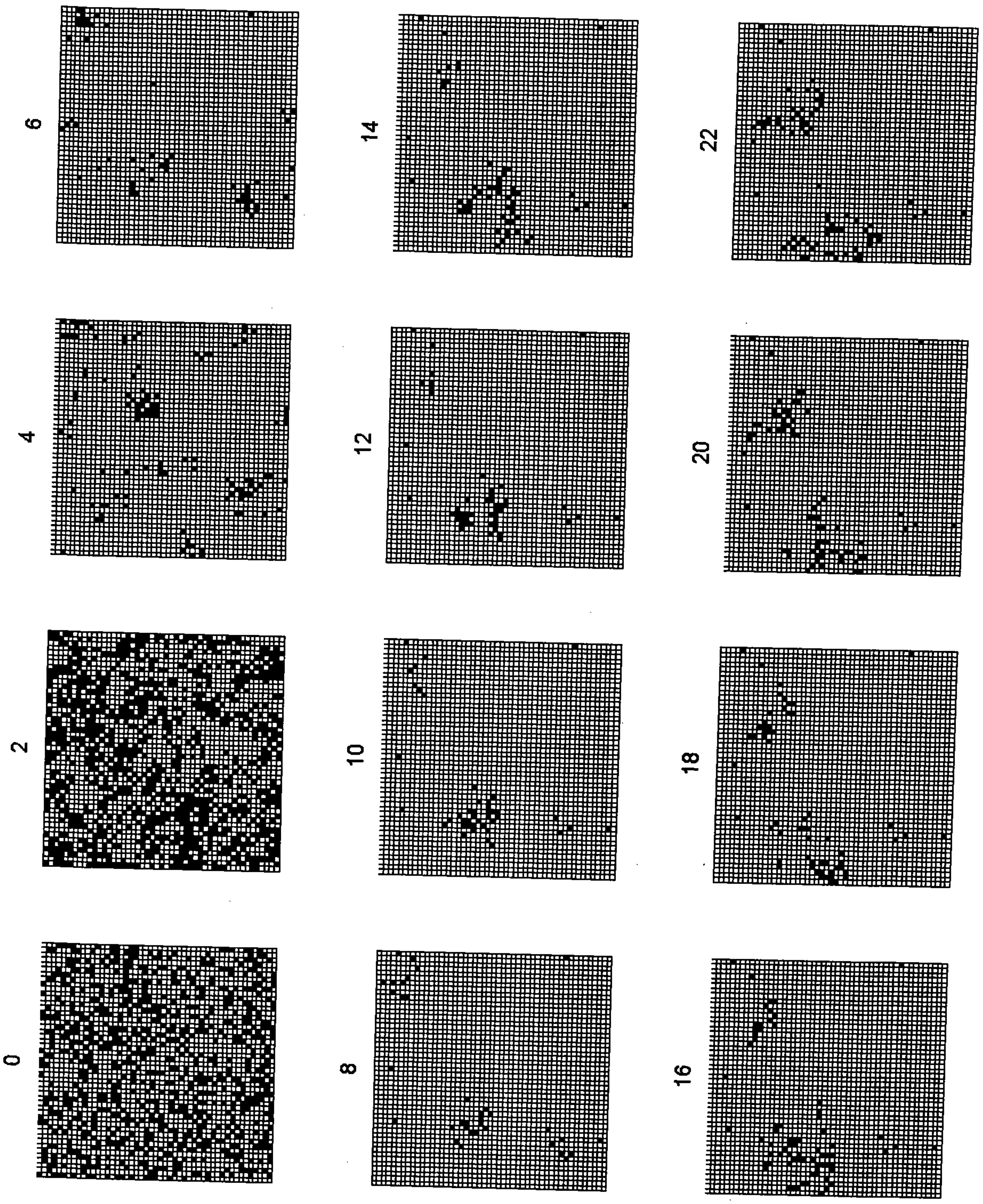


Figure 7.2(d): Population dynamics in space and time for herbivorous species #5006 in AnimSim4.

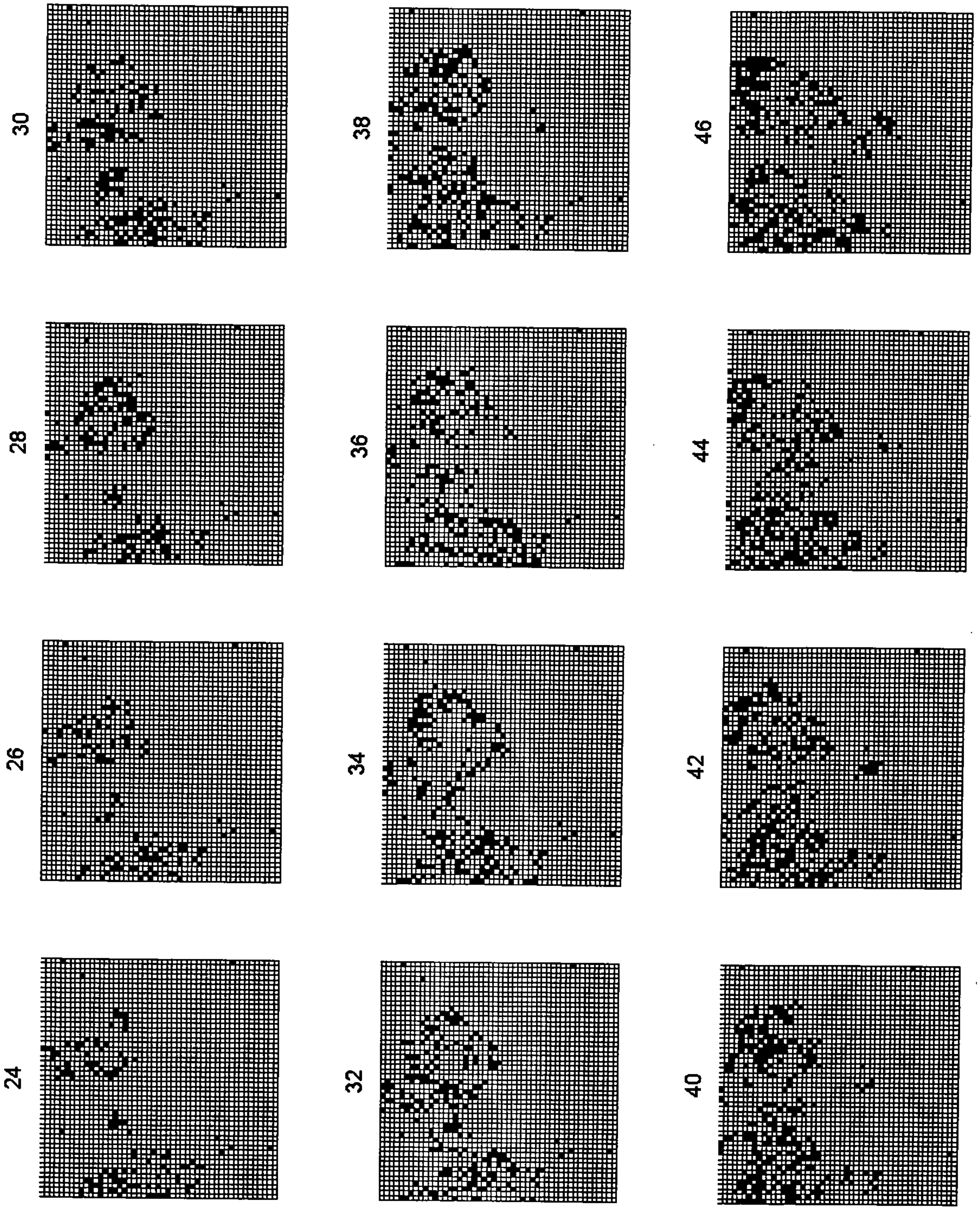


Figure 7.2(d) cont.

mass) of each species in each grid cell at the end of the growth period (late summer) is plotted for the duration of the simulation, starting at Year 0. Colouring of the squares indicates the presence of the species; darker squares depict higher masses.

In the first few years of AnimSim2, the perennial grass species (Figure 7.1(a)) is dramatically decreased in numbers due to herbivory, and is then seen to regenerate and spread over the terrain once again (note that the terrain is continuous in one dimension, thus, there is wrapping from left to right). The animal species (Figures 7.1(c) and (d)) undergo a similar initial population crash, yet do not recover. In contrast, the tree species (Figure 7.1(b)) remains relatively unaffected by the dynamics of the other species in the system. Thus, in this simulation, several species persist, with the grass species showing a remarkable ability to re-establish itself after its presence has been severely reduced by grazing. The system as a whole persists, in a state that is based largely on primary production, with only a few herbivores present at the end. In this case, the system, therefore, has settled into a meta-stable state that is qualitatively different in composition from the (unstable) two trophic-level initial state.

In contrast with the situation in AnimSim2, in AnimSim4, the two herbivorous species (Figures 7.2(c) and (d)) do re-establish themselves after their initial decreases in population size. The prevailing grass species (Figure 7.2(a)) behaves very much as the one in AnimSim2, also re-establishing its population, and the tree species (Figure 7.2(b)) remains relatively unaffected again. Thus, in this case, the plant and animal species, and the ecosystem as a whole, exhibit the autopoietic ability to regenerate, with the ecosystem evolving to a two trophic-level meta-stable state that is qualitatively similar in composition to its initial state.

### *7.1.2. Spatial self-organisation: Emergent patterns of species distribution*

Spatial self-organisation, i.e., the development of spatial patterns in the ecosystem, was observed in many instances. This phenomenon was discussed in

Chapter 5 (Figures 5.9-5.11), with reference to the distribution of plant species over the terrain, particularly the impact of herbivory on spatial distribution. Emergent spatial patterns were also exhibited by the grass species in AnimSim4 (Figure 7.2(a)), and AnimSim1 (Figure 7.3) which were subject to grazing over many years. In both cases, the grasses withstood initial, very high levels of grazing due to which their populations were decimated. In the subsequent 10-15 years, the relative abundance (in terms of mass) and the number of instances in these populations increased dramatically. This increase in mass then resulted in a corresponding recovery of the herbivorous species that consumed the grass. Thus, in the final 20-25 years of each simulation, the grasses remain distributed in a heterogeneous fashion about the terrain, but their relative abundance is lower than it was before the revival of the herbivorous species (as indicated by the prevalence of lighter coloured squares in the second halves of Figures 7.2(a) and 7.3). Thus, in both cases, a heterogeneous distribution is seen to have emerged from an initial homogeneous distribution. These patterns were persistent, and also entirely emergent, i.e., their development was not pre-specified, nor was it readily deducible from the initial model specification.

### *7.1.3. Temporal self-organisation: Power law scaling and emergent higher level dynamics*

In this section, the temporal self-organisation of the system is discussed and illustrated, through an analysis of the time series that describe the histories of the various mass forms in the ecosystem. Some of these time series have been reported in previous chapters. For example, in Chapter 4, the histories of the masses of the ecosystem sections, and of the total masses of the various compounds in the system, were plotted for the three Baseline simulations. Similar histories were shown in Appendix II of Chapter 6, for AnimSim1, and the results were compared with the Baseline simulations. In all of these cases, patterns emerged in the higher level dynamics of the system. For example, annual cycles were noted, and, in the case of AnimSim1, longer term, multi-year cycles were observed. Trends in the accumulation, maintenance, and

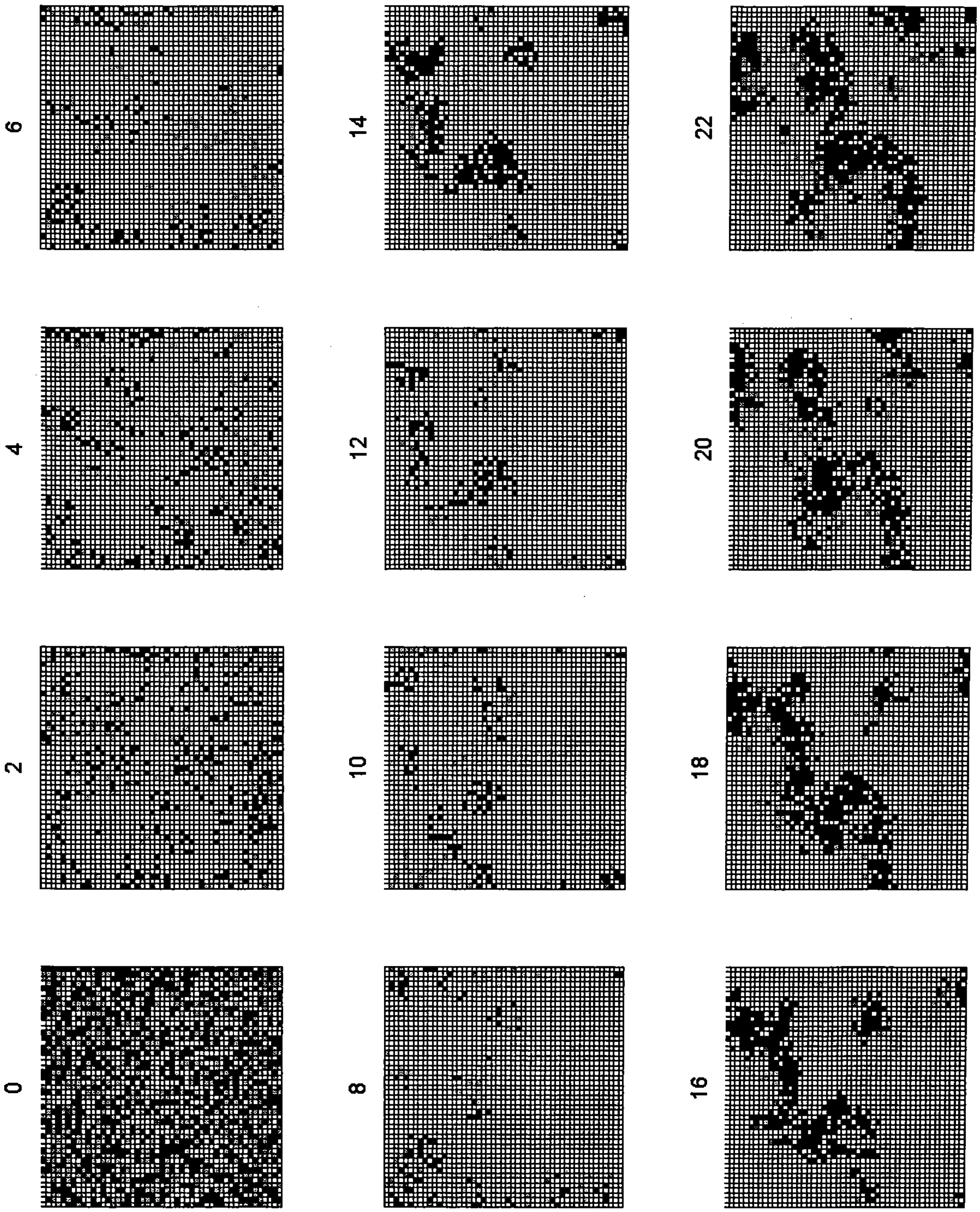


Figure 7.3: Population dynamics in space and time for perennial grass species #5146 in AnimSim1.

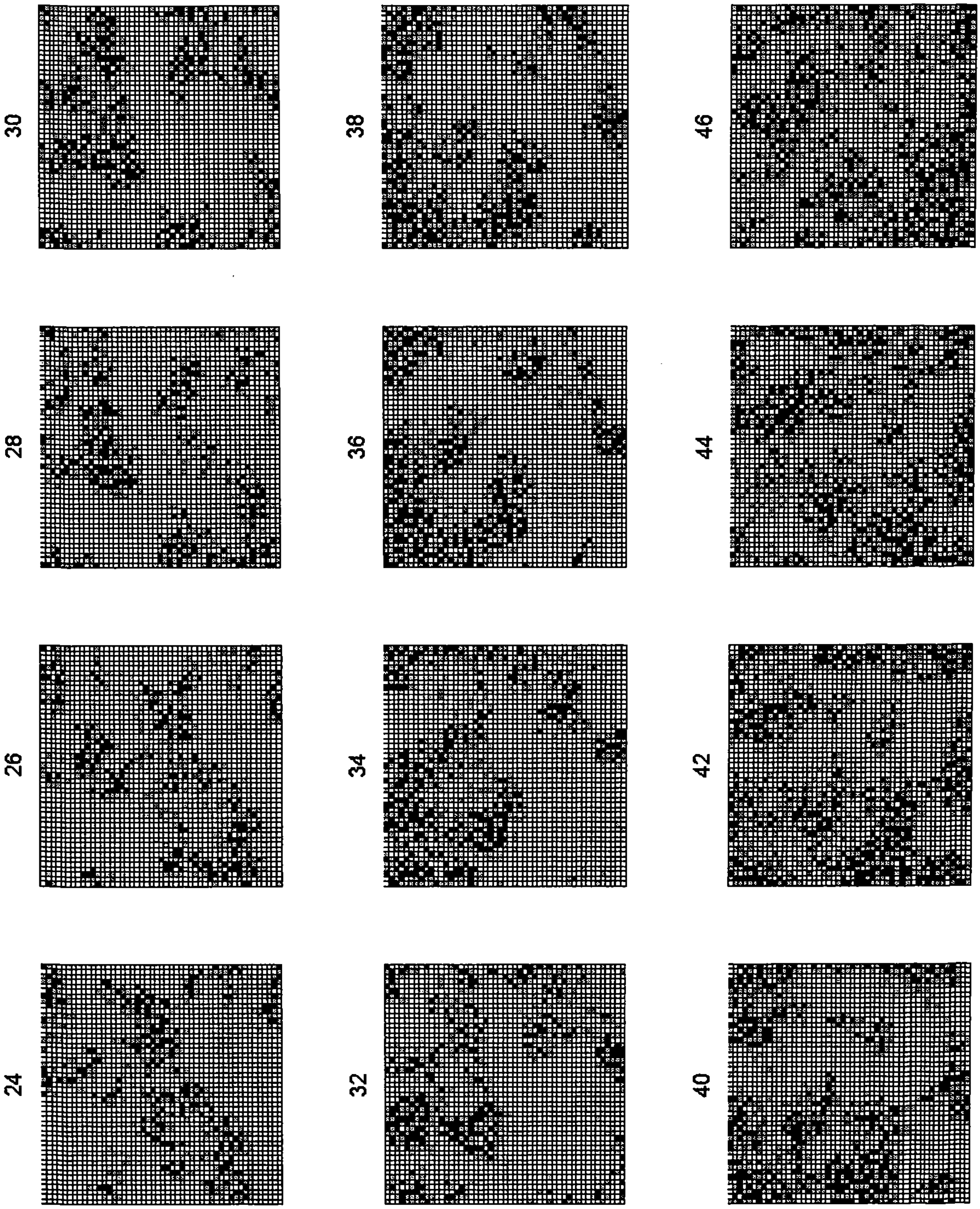


Figure 7.3 cont.

degradation of certain compounds in the system were also discerned. All of these patterns constitute a certain degree of temporal self-organisation of the system, in which the ecosystem is seen to evolve, after a transition period, to a relatively persistent state that exhibits underlying patterns, the presence of which are independent of initial conditions. In addition, as described below, a closer analysis of one aspect of these time series for four different simulations yields convincing evidence for the presence of characteristic power law scaling in the frequency domain.

The concept that "something is happening at all scales" is central to the study of complex systems, since it is the presence of multiple scale phenomena that contributes to the effective complexity of a system's dynamics. The degree to which the dynamics of a system arise due to phenomena occurring at all scales is usually studied through an analysis of the power spectrum (the squared magnitude of the Fourier transform) of a representative time series. In many natural systems, the relationship between the magnitude of a particular phenomenon (as measured by the power spectrum) and the frequency of its occurrence at each magnitude, can be described by an exponential model (a "power law") of the form:

$$f(x) = c_1 x^{-\alpha} \tag{1}$$

where $c_1$ and $\alpha$ are constants (Schroeder, 1991). In logarithmic form, this becomes:

$$\log_{10} f(x) = c_2 - \alpha \log_{10} x \tag{2}$$

For specific types of signals, characteristic values are obtained for the exponent, $\alpha$. Analysis of completely uncorrelated white noise, for example, yields $\alpha=0$. In contrast, for Brownian noise (which is highly correlated), $\alpha=2$. Signals for which an $\alpha$ value between 0 and 2 is obtained are, therefore, intermediate between the two extremes of uncorrelated and highly correlated noise. Such signals are said to exhibit "pink" noise (Schroeder, 1991). A special case is $\alpha$=1, also called "flicker" noise. If $\alpha$=1, the system is perfectly self-similar, or fractal. In such cases, the phenomenon of interest is scale invariant; that is, it occurs at all frequencies (in time or space) (West and Shlesinger, 1990).

Bak and colleagues (Bak, 1996; Bak and Chen, 1991; Bak et al., 1988) have argued that the presence of "power law scaling" in a system (i.e., when the power spectrum data fits the type of relationship of Equations (1) and (2) reasonably well, with $\alpha$>0) is a condition indicative of self-organised criticality, thereby linking these two phenomena. Correspondingly, a wide variety of systems that display power law scaling have since been claimed to be in a self-organised critical state, including Biosphere 2 ($\alpha$=1.31; Cronise et al., 1996), the global atmosphere, as measured by $CO_2$ fluctuations ($\alpha$=2.3; Cronise et al., 1996), as well as marine environments (Dachs et al., 1996), and terrestrial ecosystems (Jorgensen et al., 1998). Whether or not power law scaling can be used as a sole indication of self-organised criticality is debatable. However, both phenomenon seem to be present in any natural, complex system.

Thus, in order to examine the degree to which the global-level dynamics of the ecosystem simulations described here might be attributable to the operation of processes at multiple scales and, perhaps, to self-organised criticality, the power spectra of a number of time series were analysed for the presence of power law scaling. Four sample simulations were used: The first two, Baseline2 and Baseline3, were selected since they exhibited long-term, persistent dynamics of plant-based multi-species ecosystems. The third and fourth, AnimSim1 and AnimSim4, were selected as being representative of the behaviour of long-term, two trophic-level systems. For each simulation, the history of the total mass of carbohydrate in the biological component realm (captured once per simulated week in data recording files) was taken as an indicative variable. This time series was selected since fluctuations in the carbohydrate content of the biological component realm reflect both the response of plants to external forcing functions, as well as the presence of animals as depicted by the effects of herbivory on vegetation (animals contain no carbohydrate). It was, therefore, seen as providing an overall indication of the biological activity in the system. This time series is shown for each of the four simulations in Figures 7.4(a), 7.5(a), 7.6(a) and 7.7(a).

Each time series was then transformed into the frequency domain by computing the power spectrum of the data, thus obtaining a measure of the magnitude of the fluctuations in the carbohydrate history at different frequencies (since data was recorded only once per week, higher frequency fluctuations were not included). A least squares line was then fit as per Equation (2), and the value of $\alpha$ was computed as the slope of this line. All computations were done with Matlab v.5.1 for Macintosh (Mathworks, Inc., 1998). The power spectra, and the linear fits are shown in Figures 7.4(b) to 7.7(b).

For both of the simulations based on ecosystems configured with only plants (Baseline2 and Baseline3), the value obtained for $\alpha$ was 1.68, whereas, for the simulations based on ecosystems configured with both plants and animals, the values obtained were very close to 2.00: 1.98 and 1.96 for AnimSim1 and AnimSim4 respectively. Based on these results, it can be concluded that all of these systems display highly correlated dynamics, which is what would be expected from a system in a self-organised state. The fact that the values obtained for $\alpha$ are not close to 1 provides reasonable proof that the system is not self-similar. As discussed in Chapter 1, while the patterns in a self-similar, or fractal, system are relatively easy to describe due to their repetition across scales, a self-dissimilar system is much more difficult to describe, and is therefore, more complex.

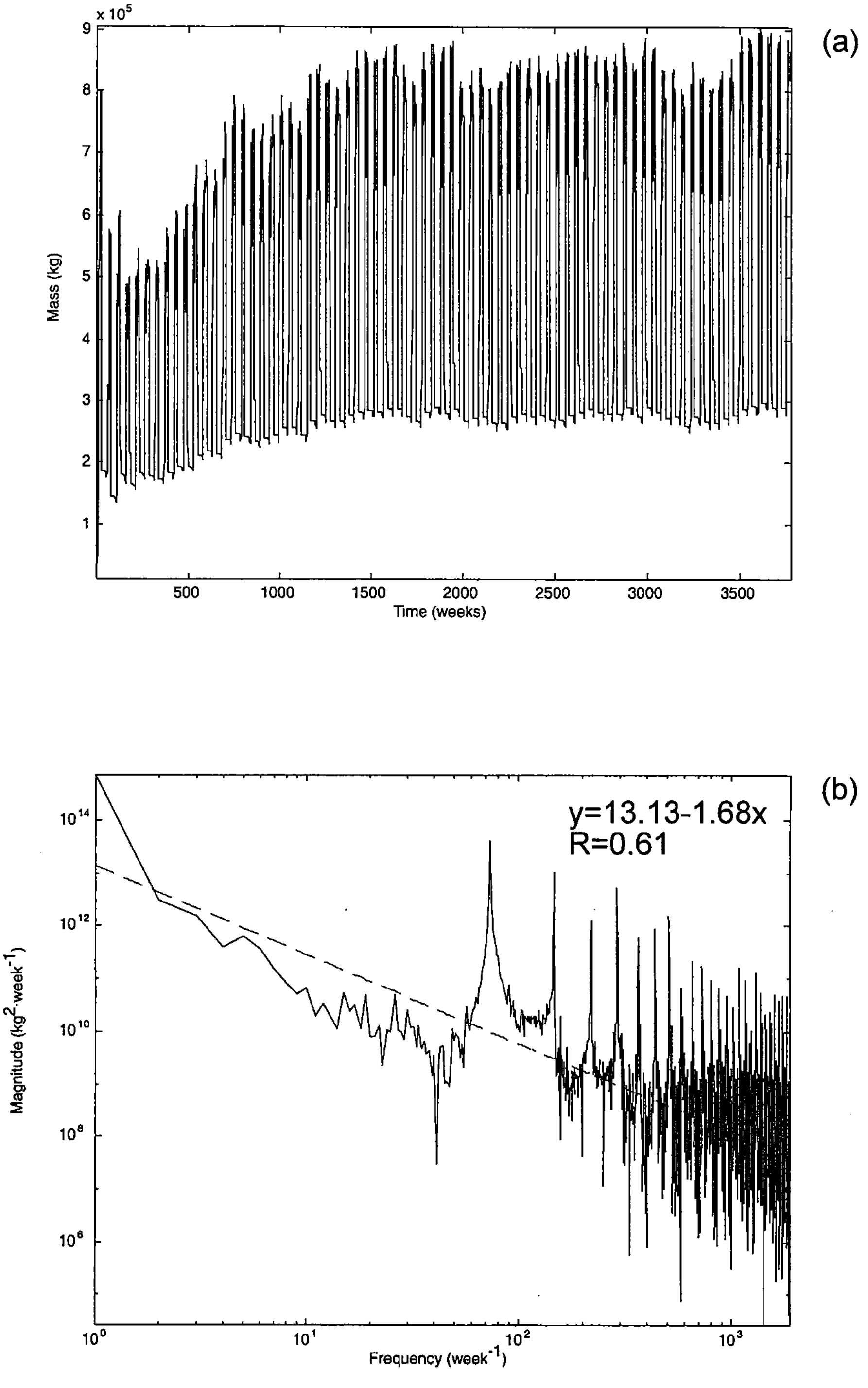


Figure 7.4: Analysis of Baseline2 for power law scaling. (a) unprocessed time series of carbohydrate in the biological component realm; (b) power spectrum of the time series.

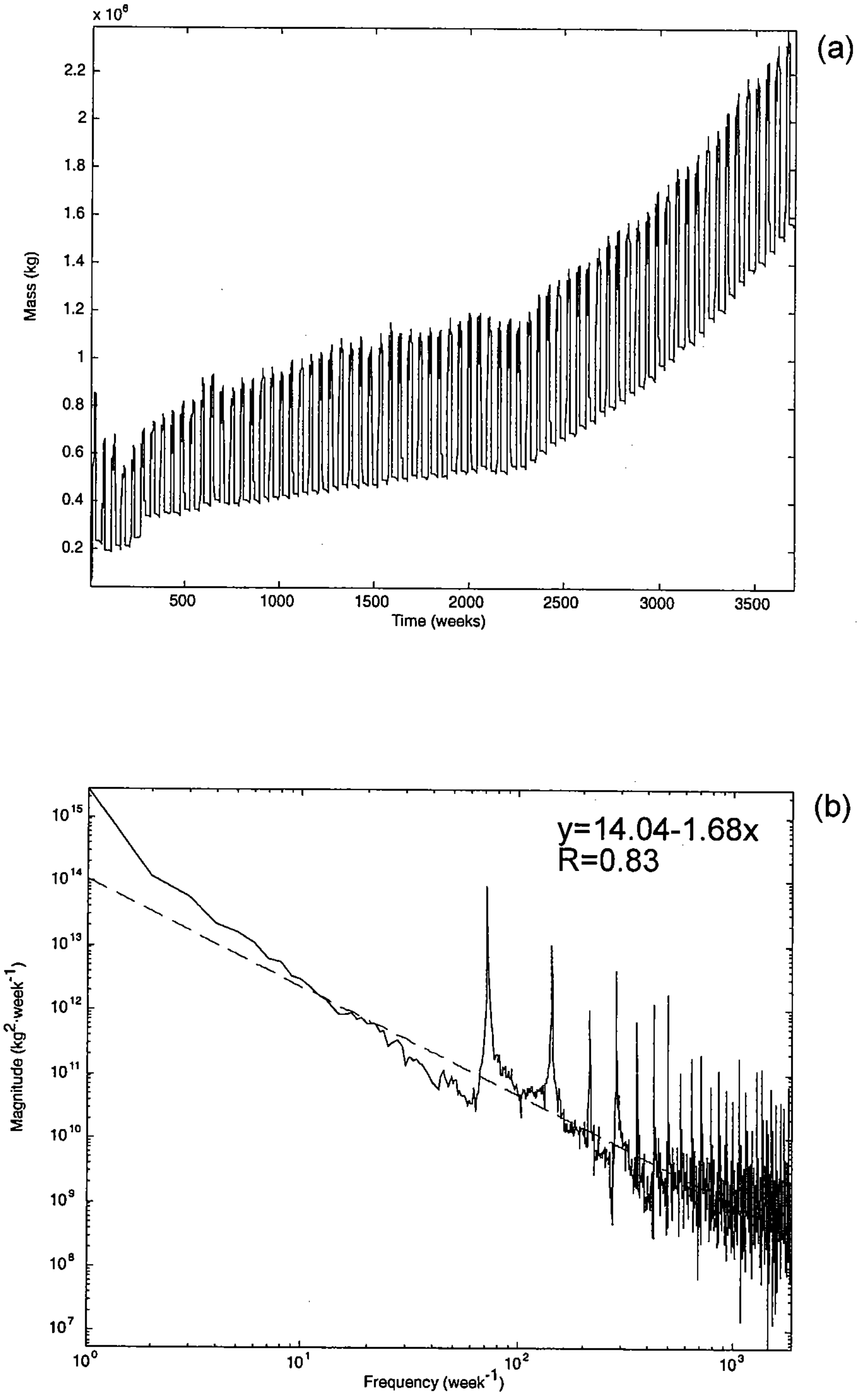


Figure 7.5: Analysis of Baseline3 for power law scaling. (a) unprocessed time series of carbohydrate in the biological component realm; (b) power spectrum of the time series.

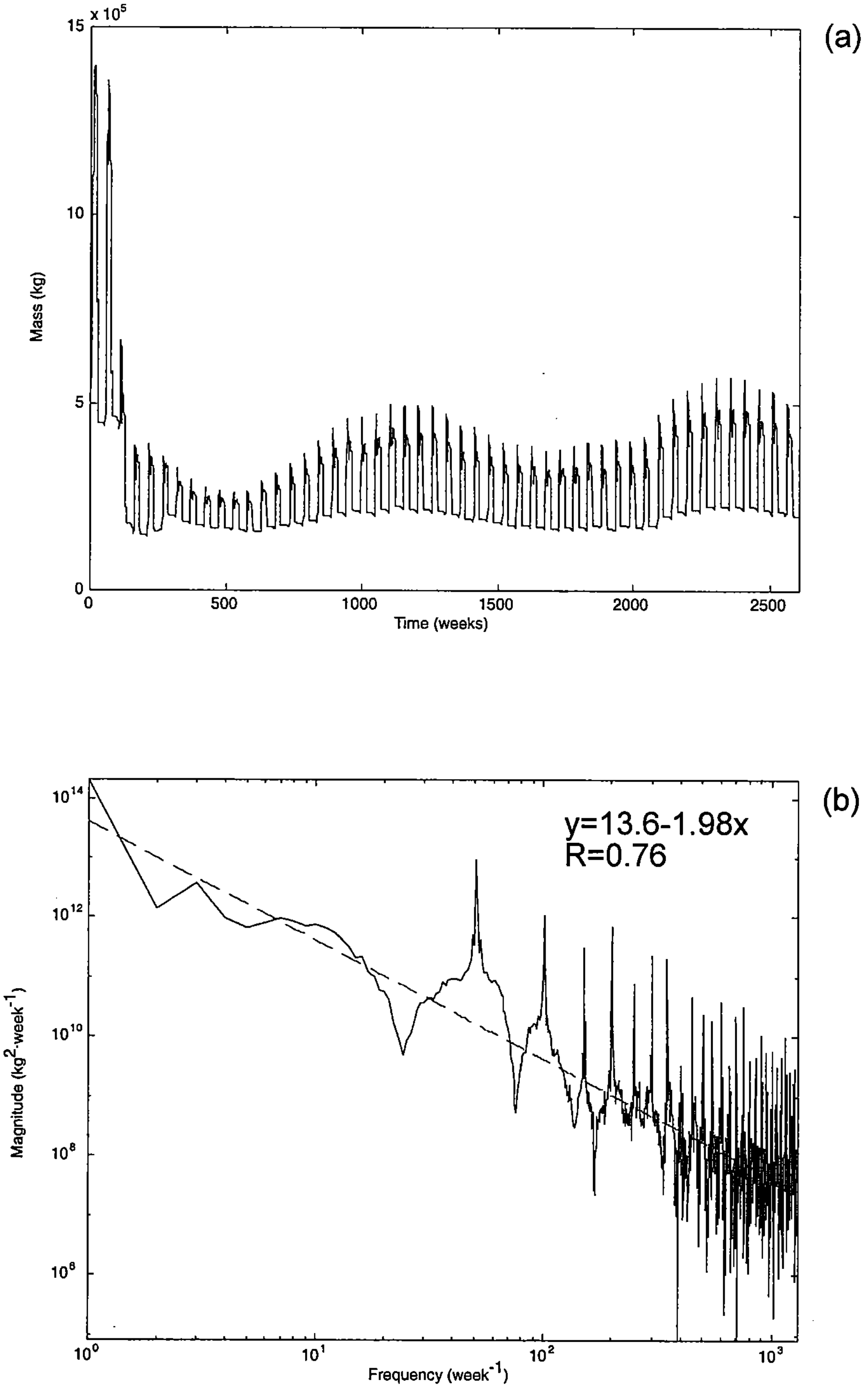


Figure 7.6: Analysis of AnimSim1 for power law scaling. (a) unprocessed time series of carbohydrate in the biological component realm; (b) power spectrum of the time series.

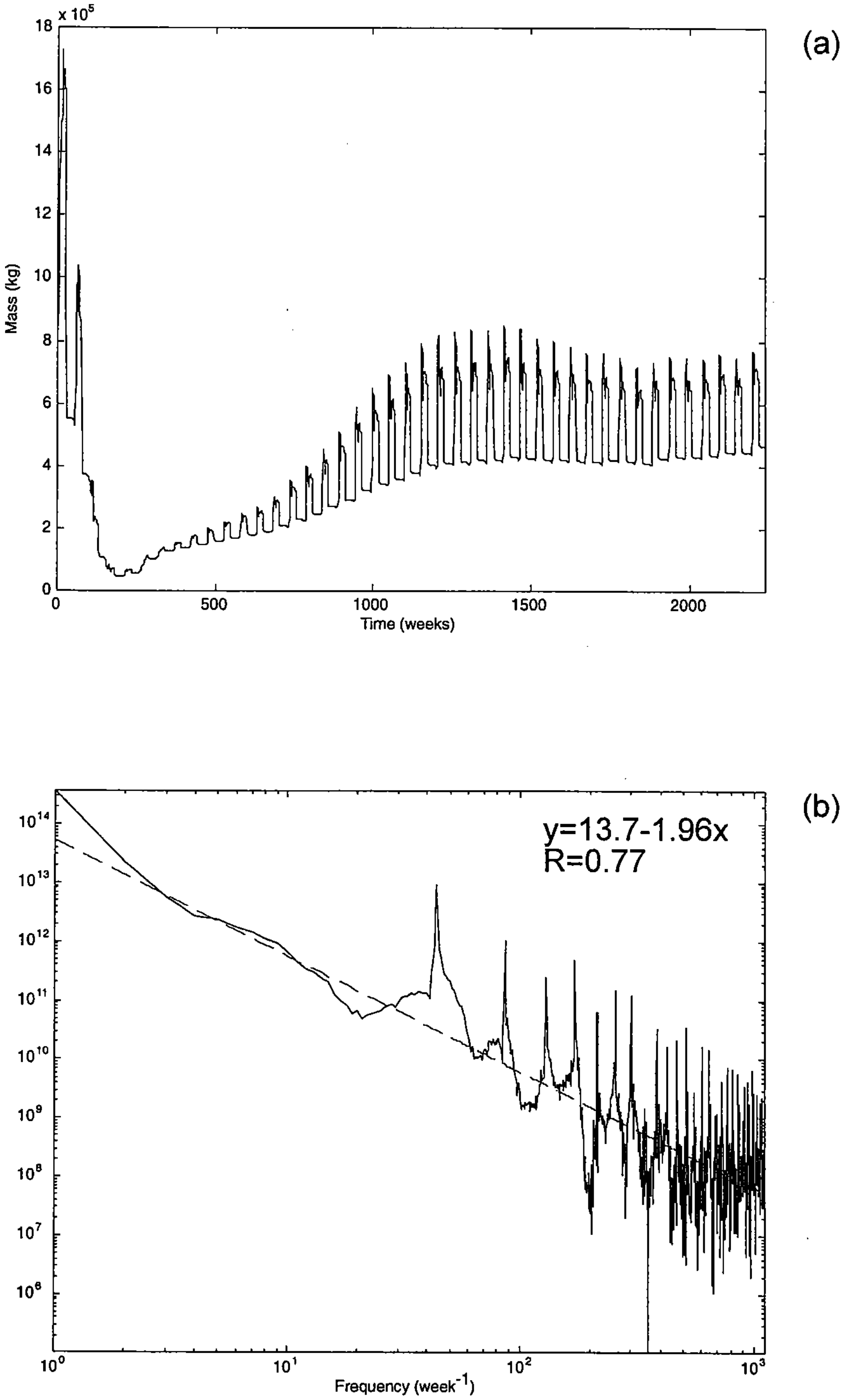


Figure 7.7: Analysis of AnimSim4 for power law scaling. (a) unprocessed time series of carbohydrate in the biological component realm; (b) power spectrum of the time series.

The value of $\alpha$ provides an interesting method of comparison of the modelled ecosystems. First, for the systems that contain animal life, and which are therefore based on a more complex food web, the values obtained for $\alpha$ were closer to 2 than the values obtained for the plant-based ecosystems. Second, systems that had similarly complex food webs, yet different biological compositions, had similar values of $\alpha$. The ecosystem of the Baseline2 simulation, for example was configured with only herbs, and that of the Baseline3 simulation had both herbs and trees, yet the same value of $\alpha$ was obtained. Third, $\alpha$ provides a measure by which two systems having very different signals in the time domain can be compared. In each of the four simulations shown here, the time series appear to be very different, yet the relationship between frequency and the power of the fluctuations, as described by $\alpha$, reveals similarities in the structures of the signals that may not otherwise have been apparent.

### 7.2. Final remarks

In conclusion, the object-based modelling approach, as used in the development of the ecosystem model described here, has proven to be an effective way to represent ecosystems and to simulate their behaviour. The approach has enabled a relatively complete depiction of an ecosystem, including material cycling, food web relationships, and the presence of many different types of organisms. Simulation results have been achieved that exhibit phenomenological similarities to physical ecosystems (e.g., trends in biomass accumulation, and population dynamics), as well as features typical of complex systems (e.g., autopoiesis, spatial and temporal self-organisation). Overall, the approach has enabled the development of a model that is a novel contribution to the field, both with regards to the scope and resolution with which an ecosystem is represented, and the type of comportment that is elicited in simulations with the model.

The model was developed as a biosystems engineering tool, thus, many of the components and their functionality have been modelled in a relatively simplistic manner. This simplicity, especially with regards to the representation of plants and animals, is justified in the larger context, since it is the global behaviour of the system, not of the components, that is of interest to any systems engineer. In addition, this approach facilitates the inclusion of a wide variety of different types of components, in order to adequately represent all of the important processes that occur, and that should be considered in any engineering endeavour.

The results obtained demonstrate the potential flexibility and utility of the model for use in ecosystem engineering applications. The ability to configure the model through the definition and specification of the composition, functionality, and initial state of all of the components of an ecosystem considerably facilitates experimentation with various design scenarios; the space-station used for demonstration purposes in this project being just one of many possibilities that might be explored. In addition to serving as an ecosystem design tool, a model such as this one can also be used to study the effects of different management strategies on the overall comportment of an ecosystem. For example, if configured appropriately, the model could be used in simulation to investigate a number of "what if" scenarios related to ecosystem restoration, and environmental assessment projects.

The model can also be used to provide insight into the general nature of the structure and functioning of natural ecosystems, as well as for the theoretical study of the underlying relationships between constitution and comportment in many related types of complex biological networks. As illustrated with AnimSim4, the ability to modify method routines allows for the study of the effects of different animal behaviours on the survival of a species, and on the dynamics of the ecosystem as a whole. Simulations such as this one can be used to investigate a number of key issues in theoretical ecology related to food web structure, competition theory, and ecosystem resilience, for example.

For the EcoCyborg Project, the model will be used to explore the design of intelligent control systems for ecosystems. The fact that the model has been shown to exhibit complex dynamics is important in this context, since it will serve as a test system for various control strategies. The model has been shown to exhibit behaviour that, like a natural ecosystem, is somewhat predictable at the global level for short periods of time, yet is effectively unpredictable at the lower level, and over the long-term. This type of situation presents a number of challenges for any control system (Solé et al., 1999). By reproducing these dynamics to some degree with the model, it will be possible to test simulation-based control mechanisms in a relatively realistic context, before experimenting with physical systems.

As previously mentioned, the model is one of the most detailed (with respect to scope and resolution) object-based representations of an ecosystem currently in existence. When implemented in simulation, as illustrated by the results presented here, it can successfully reproduce dynamics that are both complex, and also not unlike what would be expected from a natural ecosystem. Thus, the number of objects included, and the nature of their interactions, gives rise to a highly complex, dynamic network in the virtual realm, that can be used to simulate the behaviour of ecosystems. The approach, therefore, has proven successful, and is recommended for the simulation of other similar types of biosystems, in pursuance of the complex quest of learning to engineer life.

## 7.3. References

Bak, P. 1996. *How Nature Works.* New York, NY: Springer-Verlag.

Bak, P. and K. Chen. 1991. Self-organized criticality. *Scientific American* 264(1): 46-53.

Bak, P., C. Tang and K. Wiesenfeld. 1988. Self-organized criticality. *Physical Review A* 38(1): 364-373.

Cronise, R., D. Noever and A. Brittain. 1996. Self-organized criticality in closed ecosystems: Carbon dioxide fluctuations in Biosphere 2. *International Journal of Climatology,* 16: 597-602.

Dachs, J., J. Bayona and S. Fowler. 1996. Flicker noise in vertical fluxes of particle-associated contaminants in the marine environment. *Environmental Science and Technology,* 30(11): 3392-3396.

Jørgensen, S., M. Henning, and S. Nielsen. 1998. Ecosystem as self-organizing critical systems. *Ecological Modelling* 111: 261-268.

Green, D. G. 1994. Connectivity and complexity in landscapes and ecosystems. *Pacific Conservation Biology* 1(3): 194-200.

Schroeder, M. 1991. *Fractals, Chaos, Power Laws.* New Jersey: W.H. Freeman & Co.

Solé, R., J. Gamarra, M.Ginovart and D. López. 1999. Controlling chaos in ecology: From deterministic to individual-based models. *Bulletin of Mathematical Biology,* 61: 1187-1207.

West, B. and M. Shlesinger. 1990. The noise in natural phenomena. *American Scientist,* 78: 40-45.

## Chapter 8. Contributions to knowledge

1. ***Intellectual contributions to the EcoCyborg Project:*** Although not explicitly discussed in the body of this thesis, a substantial part of this project involved the elucidation of a number of key biosystems engineering concepts within the context of the EcoCyborg Project. These included: the investigation and formulation of cyborging as a possible biosystems engineering strategy; the classification of ecocyborgs as living systems; the conceptualization of a hypothetical ecocyborg and its constituent components; and a methodology and approach for biosystems design in the context of complex system studies. (These concepts are presented in Appendix C.)

2. ***Elucidation of the biosystem modelling process in terms of three stages:*** conceptual, representational, and computational, with reference to ecosystems as a case study. Also: conceptualization of an ecosystem as a complex system, and of the specific changes that arise in the modelling process as a result of the use of this conceptual model. (Chapter 2)

3. ***Introduction and development of the term "object-based"*** to describe a class of representational models with which a system is depicted as a collection of many interacting components (with states described with attributes, or properties, and functionality described with rule-based expressions). Interpretation of current ecosystem modelling literature within this context. (Chapter 2)

4. ***An approach for the creation of a generally configurable model that can be used to explore the engineering of ecosystems.*** (Chapter 3)

5. ***A methodology for object-based ecosystem modelling,*** in which a model is configured in separate definition and specification stages, thereby allowing for complete flexibility in terms of the creation of object types and the specification of their initial states. (Chapter 3)

6. ***A framework for the modelling and simulation of ecosystems according to the above methodology.*** Specifically: a model configuration process that involves the use of configuration programs with which object attributes and methods can be listed, hierarchies of classes that use these can be defined, and species definitions can be composed, based on the class definitions. Also: definition of a food web in terms of species food preferences and the specification of this within the framework. And: use of specification programs with which the initial states of the material storage, encompassment, and biological component realms of an ecosystem model can be set. (Chapter 3)

7. ***Elucidation of the initial value problem encountered when attempting to specify the states of biological components for a fully-formed ecosystem, and the recognition of this as a generally applicable problem in biosystems engineering.*** This problem was treated in the development of an object-based model, with the use of attribute creation values that specify distributions of the values within a population, for both mature and juvenile instances. (Chapter 3)

8. ***A generally configurable ecosystem model that demonstrates the use of the methodology described above, and the subsequent implementation of this model in simulation for engineering research purposes.*** (Chapter 4)

9. ***A generic, object-based plant model for use in ecosystem simulations.*** Specifically: an object-based model that can be configured to represent many different plant types, at a high level of resolution, and the demonstration of the effectiveness of the model in time driven simulations with small time increments. (Chapter 5)

10. ***A generic, object-based animal model for use in ecosystem simulations.*** Specifically: an object-based model that includes representations of both an animal's behaviour and it's physiological state; illustration of the use of the model in simulation, for animals at two different trophic levels. (Chapter 6)

11. ***Investigation and confirmation of the applicability of object-based modelling for the representation of ecosystems to a greater degree of detail, and with a wider scope, or extent of treatment, than has been previously achieved with an object-based model.*** The applicability has been demonstrated via: the achievement of simulation results for plant-based ecosystems that predicted specific biomass levels that are similar to those found in physical ecosystems (Chapter 5); the demonstration of the effects of herbivory on the spatial variation of plant species (Chapter 5); and the demonstration of multi-trophic level population dynamics, including predator-prey dynamics. (Chapter 6)

12. ***Presentation and analysis of simulation results based on an object-based ecosystem model, that exhibit complex comportment similar to that of natural ecosystems, including autopoiesis and spatial and temporal self-organisation.*** (Chapter 7)

# Chapter 9. Recommendations for future work

Since the EcoCyborg Project is ongoing, and the ecosystem model developed as part of the work described will be used in future simulations, a number of recommendations for its improvement are presented here. Many of these are based on ideas that arose as afterthoughts during development and testing, and which inevitably come about as the result of trial and experience.

The are two kinds of recommendations presented here. The first involve specific improvements that could be made to the model. The second address many of the problems that were encountered due to hardware limitations, some of which are likely to be overcome as computing capacity increases. Thus, the model was, in a sense, designed to grow into the future. Of course, as the model is improved, and more details are included, the computing power required for simulations will increase correspondingly, causing further hardware limitations.

## 9.1. Suggested improvements to the ecosystem model

### *9.1.1. Make soil decomposition rates temperature and moisture dependent*

Decomposition rates for decomposing plant and animal matter, and for soil organic matter, are currently constant for any given ecosystem configuration. It is recommended that these rates be made temperature and moisture dependent to better reflect the effects of weather and environmental conditions on the accumulation of these materials in the soil.

### *9.1.2. Make trees into hybrid lump/individual objects*

Due to the vast number of seeds created by a single mature tree, the populations of these species very quickly reach the tens of thousands. Most of these tree objects are recently germinated, small trees that are competing for dominance on the forest floor. To reduce the computational burden caused by the presence of so many trees, it would make sense to treat the vast majority of them as lumps, with a maximum of one instance per grid cell, in a manner similar to herbaceous species. Individual tree objects could then be allowed to emerge from the lump tree objects when a mature tree in that location dies, allowing for a new tree to take its place.

### *9.1.3. Plant root depth*

Currently, plant root depth is not considered at all. This means that as long as there is some available water in a given grid cell, a plant is assumed to be able to absorb it. It is recommended that, in future versions of the model, plant root depth be included as an attribute of the plant objects and that this value be considered in the plant water absorption method. Doing so would allow the study of plant distribution patterns on the terrain as a result of drought or unequal rainfall accumulation.

### *9.1.4. Improve the animal energetics methods*

The current version of the model does not take into account heat generation or thermoregulation in animal energetics. The inclusion of these aspects would make the representation of animal objects, particularly their respiration requirements, more realistic.

### *9.1.5. Add history dependent decision making to animal behaviour*

Animals currently act according to simple if-then decision rules in which they examine their current state and their surrounding environment, and then select an activity. This decision is not based on the animal's activity in the previous time interval. Thus, if an animal is fleeing from a prey, or moving towards the pond to drink, it will not necessarily continue this activity in the subsequent time increment (unless that activity is still dictated by the animal's current state and/or the surrounding environment). It would be useful for an animal to "know" what it had been doing previously, so that then a decision to continue that activity could be made prior to selecting another, unrelated task.

### *9.1.6. Add inheritance and evolution of traits*

Inheritance and evolution were not included in the model since, for most species, the noticeable effects would occur on a time scale much longer than that of most simulations. If, however, the execution speed of simulations were to increase substantially in the future, it would be interesting to incorporate an evolutionary/adaptive aspect in the model whereby, minimally, new instances would inherit the traits of their progenitors.

### *9.1.7. Use of multiple time increments*

To reduce the computational intensity of the model when used in simulation, different types of objects could be updated according to different time increments. Although some objects types such as animals, are best simulated with a short (i.e., 10 minute) time increment in order that certain behaviours be represented, for other types of objects, a longer time increment might be used. Many of the terrain functions such as decomposition, for example, could probably be carried out much less frequently without any discernible effect on the overall system dynamics. These functions could be called once every 10 or 20 simulation cycles, instead of every cycle. Thus, it is recommended that objects be updated according to a staggered scheme, so that only those objects that really need updating are activated every cycle.

## **9.2. Suggested improvements to the ecosystem configuration programs**

### *9.2.1. Create species attribute value testing modules*

As previously mentioned, one of the greatest shortcomings of the object-based modelling approach when organisms are represented at the individual level, is the selection of appropriate attribute values and the amount of time required to input these. Through the use of the biological component class hierarchy, in which standard classes were defined with attribute creation values that were inherited by species classes, some of the time required for model specification was reduced. Still, it was often difficult to specify appropriate values at the species level for certain key attributes, usually those related to the growth and maintenance of organisms. It would, therefore, be useful to develop several simple testing modules in which "sample" organisms could be simulated. With this, instances of a new species could be simulated in isolation, in order to assure the appropriateness of the chosen attribute values. An animal testing module, for example, would allow an animal to grow in the presence of an unlimited, generic food source, without any other inputs, so that the values of attributes such as the metabolic rate and growth constants, stomach size, digestion rate, and fat accumulation rate might be checked. A beta version of such a plant testing module has already been developed, in which a plant can be grown with unlimited energy, nutrient, and water availability in order to check its seasonal and lifetime growth patterns. This module could be improved and implemented in conjunction with test modules written for other types of species.

### *9.2.2. Migrate the definition and specification programs*

The ecosystem definition and specification programs are currently run under IBM's OS/2 (they were written at a time when the project was being developed for the OS/2 platform, and simulations were being executed under a multitasking "Simulation Workbench" designed to take advantage of the shared memory allocation and multi-session capabilities of this operating system[1]. This is obviously no longer a viable platform, and it is therefore recommended that these programs be migrated to a more universally available platform such as Windows 98 or Linux. In addition, these programs were written to use a set of text-based user-input subroutines developed by a previous member of the research team and written in BASIC. As part of the migration process, it is recommended that the programs be re-written to take advantage of a modern, graphical user interface.

---

[1] Lacroix, R., R. Kok, and O.G. Clark. 1996. Use of a multitasking operating system as a setting for the simulation of an enclosed agro-ecosystem under cognitive control. *Canadian Agricultural Engineering* 38(2):129-138.

## 9.3. Suggested improvements to the simulation program

### *9.3.1. Enable system initialisation in stages*

Initialization of the ecosystem in a fully formed state presented a number of logistical difficulties (i.e., how to specify the state of a mature tree without first simulating its life history; how to specify an appropriately large colony of small herbivores to support a carnivorous species; etc.). In order to avoid some of these difficulties, and to reduce the amplitude of system fluctuations during the transitory period, a staggered startup approach, whereby the different trophic levels would be introduced incrementally, might be more appropriate. With such an approach, the vegetation could be introduced first, and be allowed to grow and reach a climax state before the introduction of species of higher trophic levels. The number of startup stages required in each case would likely be related to the intricacy of the ecosystem's constitution. This staggered startup method would be closer to the approach that would likely be followed in the construction of a similar physical system.

### *9.3.2. Enable simulation restarts*

Simulations based on the ecosystem model take quite long to execute, requiring many days of machine time on current desktop computers. Since power outages and system crashes seem to be inevitable, it would be valuable to be able to "restart" a simulation in mid-run, rather than beginning anew. At present, not enough data is saved to accomplish this, and a "restart" function has not been written into the code. It is recommended that the entire system state be saved to disk regularly, perhaps once per simulated year, so that a stopped simulation can be recommenced at exactly the point where it had left off. This requires saving not just the current state of the ecosystem, but also of the random number generator, so that the system can be restarted exactly as it was, creating a reproducible experiment (i.e., the final results should be identical to those had the simulation never been stopped).

### *9.3.3. Enable simulation pauses*

For reasons similar to those expressed in favour of a restart option, a simulation "pause" function that temporarily spooled the process to disk would be useful. This would allow a simulation to be momentarily placed on hold so that the host machine could be accessed for other purposes. This would allow the simulation to be run on multipurpose computers (e.g., those in a laboratory or office) in a way such that it could make use of idle time, yet be paused by a user who wished to garner all of the hardware resources for some other application.

### *9.3.4. Enable simulation restarts with different control schemes*

For future experimental work, it would be very useful to be able to save a mature, non-transient, system state, and then be able to run multiple experiments with that state as the initial condition. In order for the experiments to be different, a more sophisticated control system would have to be included, which could then be used to implement different types of management decisions, or to impose fabricated disturbances. For example, it would be interesting to run parallel simulations, and to cause a severe drought to occur in one, or to cut down all of the mature trees in another, etc., and to then observe the divergence between them.

## 9.4. Suggested improvements to the simulation environment

### *9.4.1. Create a real-time graphical display of simulations in progress*

Although it is not necessary, a real-time, graphical display would make it easier to monitor the progress of a simulation. One approach to this would be to create an interactive display, in which human observers could issue control decisions which would then be implemented during the simulation via special effector mechanisms. To avoid increasing the overall time required to run a simulation, a display program should probably not be run on the same computer as the one on which the simulation is executing.

### *9.4.2. Distributed processing*

Many of the recommendations discussed above were not implemented due to adequate hardware not being available on which to run simulations. Considering the high computational intensity of the model, and the anticipated increase in computational requirements should further work be undertaken, alternative processing schemes will have to be used. At this point, the possibility that appears to be most promising with regards to potential increases in processing capacity, would be to divide the task amongst multiple, networked computers. Thus, it is recommended that work be done to implement the model in a distributed fashion in which a central simulation manager would handle the task of dividing and allocating the work load among CPUs, and manage the data sharing between machines.

## 9.5. Recommended future experiments

### *9.5.1. Experiments with larger scale ecosystems*

The size of the terrain in all of the experiments run to date was only 500m x 500m. This size was selected in order that the landscape could be represented at a reasonably high level of resolution (i.e., 10m x 10m cells, and even these are considerably larger than most of the organisms that occupy them) without requiring excessive amounts of computer memory and processing resources during a simulation. This terrain proved, however, to be too small to support large carnivorous or herbivorous species without external inputs. Thus, it would be very interesting to run future simulations with ecosystems configured with much larger terrains that could support a greater variety of species. This would facilitate experimentation with different ecosystem compositions, allowing more detailed analysis of the structure and stability of food webs.

### *9.5.2. Experiments with more species*

The model has been developed to support up to 1000 different species. Several hundred species classes have been defined, but only a handful of these were tested, due to the time required to fully specify and test the attribute values of the species with sample instances (see recommendation above re: the development of species attribute value testing modules). Future work should involve further testing of species definitions, and the configuration of ecosystems with many more species.

### *9.5.3. Experiments with different forcing function series*

All of the experiments performed to date have been undertaken with the same forcing function series, i.e., every simulation was configured with the same weather generator parameter values (see Chapter 7). It would be interesting to explore the system response to forcing functions that were substantially different from those used, and to perturb the system with weather related disturbances such as extended dry periods or exceptionally hot weather.

# Appendix A. The Weather Generator: Description of the weather routines used to generate forcing function values for the ecosystem model.

## A.1. Introduction

In this appendix, a brief description of the Weather Generator is given. The Weather Generator is used to supply forcing function values to the ecosystem model. It consists of a collection of routines that are called by the ecosystem simulation program to calculate temperature, radiation and rainfall values for each time increment. The source code for these routines is in the file "weather.c" on the enclosed CD-ROM (Appendix B).

## A.2. Objectives

The Weather Generator was created to meet the following general requirements:

- it must be able to compute a temperature (°C), radiation intensity ($W \cdot m^{-2}$) and rainfall rate ($m \cdot s^{-1}$) value for any time of the day, any time of the year

- the series of values produced must be reasonably similar to those observed for terrestrial ecosystems, e.g., with regards to the annual and daily patterns, variability, etc.

- it must be configurable so that a variety of different climate types can be simulated

## A.3. Method

Three independent streams of data are generated. That is, rainfall, temperature and radiation values are produced that are not cross-correlated (the implication of this is that it may be sunny and raining at the same time).

### *A.3.1. Temperature*

Series of daily average temperature values are generated, a year at a time, using the method described in Parrott et al. (1996). This method is based on a Fourier analysis of physical temperature data, and the subsequent reconstitution of this data as a summation of sinusoids and high frequency "noise" values. Corresponding series of minimum daily temperatures and maximum daily temperatures are also generated with this method, except that "noise" values are omitted. Different sets of daily average, minimum and maximum temperatures are generated for each year of simulation. These are computed "on demand" as a simulation proceeds (i.e., for a fifty year simulation, the model does not produce 50 years of temperature data at the start). At any time during a simulation, however, the values for the previous, current and subsequent year are stored in memory.

To obtain daily time series of temperature values, a polynomial curve is fit through the daily average, minimum and maximum temperatures for three days in sequence, where the current day is always the second in the series. For the purposes of the curve fit, it is assumed that the daily average value occurs twice per day, at 8h and 22h, and that the daily minimum occurs at 5h and the daily maximum occurs at 14h. This results in a continuous series of temperature values for the current day (the previous and next days are only used to smooth the polynomial curve in order to avoid large discontinuities in temperature between days). Thus, when a temperature value is requested by the simulation program (currently at intervals of 10min), the value is calculated by selecting the appropriate point on the daily temperature curve.

The temperature routines require a number of parameter values. A list of the parameters, with descriptions, and the values currently used, is presented in Table A.1. By selecting the appropriate parameter values, the temperature model can be used to generate a wide variety of different temperature regimes that mimic any location on Earth. See Parrott et al. (1996) for examples.

### *A.3.2. Radiation*

Radiation intensity ($W \cdot m^{-2}$) values are first generated with standard equations to predict the amount of solar energy received by a horizontal surface on Earth at a given latitude, neglecting atmospheric influences (i.e., water vapour and local barometric pressure). These equations are based entirely on the predictable behaviour of the Earth-Sun system and can be obtained from any elementary text on the subject. Next, to introduce some variability to the climate, and to attenuate the theoretical values somewhat, a stream of autocorrelated hourly "noise" is calculated for each year, and predicted radiation values are multiplied by their corresponding hourly noise value. A brief description of the procedure follows.

The theoretical radiation intensity (*I*) for a given increment of time between sunrise and sunset is calculated as (Christie, 1987, Table 7, Eqn. 8):

$$I = \frac{S}{r^2} p$$

where,

S = solar constant, $W \cdot m^{-2}$

*r* = the radial distance of the Sun from the Earth (expressed as a fraction of the mean radius; Christie, 1987, Table 7, Eqn. 14 & Table 8)

and,

$$p = \sin(\phi)\sin(\delta) + \cos(\phi)\cos(\delta)\cos(h)$$

where,

$\phi$ = latitude on Earth, radians

*h* = angular distance of the sun from the meridian, radians (dependent upon the time of day)

$\delta$ = declination of the sun, radians
(dependent upon the time of year)

The time of sunrise and sunset for each day of the year at a given latitude are also obtained from standard equations. Radiation intensity is zero between sunset and sunrise.

The above equations give an accurate prediction of the amount of sunlight reaching the Earth's surface at a given latitude and time of year in the absence of atmospheric effects. The actual radiation intensity received at the Earth's surface is usually reduced by 20-50% due to the presence of water vapour, gas, and dust particles in the atmosphere. In most other radiation models, equations are used to predict this effect based on humidity and air pressure, as well as a number of empirical constants that describe the effects of these on an atmosphere which has the thickness of the Earth's. Instead, with the Weather Generator, this effect is simply approximated by multiplying the theoretical radiation values by an autocorrelated "noise". Thus, for each year, a series of 8760 (hourly) autocorrelated numbers are created, whose values range between 0 and 1. For each hour of the day, a theoretical radiation intensity is calculated as described above, and this value is then multiplied by the noise value for that hour. It is assumed that the value of $I$ varies in a continuous, linear fashion from one hour to the next, thus if a value of $I$ is required for an intermediate time, it is computed with a linear equation that is fit between the two surrounding hourly points.

The behaviour of the radiation model can be adjusted by altering the values of the latitude variable, the solar constant, and the noise autocorrelation. These are all input as simulation parameter values to the Weather Generator (Table A.1).

### *A.3.3. Rainfall*

The rainfall rate ($m \cdot s^{-1}$) is determined stochastically. Four parameters are used to compute this: (1) the average rainfall rate; (2) the standard deviation of the rainfall rate; (3) the probability of rain; (4) the probability of rain if it was raining during the previous time increment. First, to determine whether or not it is raining at a given time, a uniformly distributed random number between 0 and 1 is computed and its value is compared with the appropriate rainfall probability. If the value is lower than the probability of rain, it is raining. A rainfall rate is then computed by selecting a value randomly from a set of normally distributed numbers having the mean and standard deviation specified as parameter values. To keep the rainfall rate within realistic limits, this distribution is truncated at a minimum of 0 $mm \cdot hr^{-1}$ and a maximum of 25 $mm \cdot hr^{-1}$ (6.94E-06 $m \cdot s^{-1}$). In this model, the rainfall rate, and probabilities of rain, do not change with seasons.

## A.4. Configuration of the Weather Generator for use in ecosystem simulations

The Weather Generator was configured to produce a mild climate for the EcoCyborg Ecosystem Model. The parameter values used for all simulations run to date are given in Table A.1.

## A.5. Model performance

A typical week of temperature, radiation and rainfall values produced by the Weather Generator is shown in Figure A.1. As well, three typical years of temperature, radiation and rainfall are shown in Figures A.2, A.3 and A.4, respectively. From these, it is evident that the results fall within the ranges specified by the parameter values, while at the same time being somewhat different from one year to the next. Some 50-year statistics are given in Table A.2.

|  |  |  |
|---|---|---|
| **Daily Average Temperatures* (°C)** |  |  |
| a0 | Mean | 21 |
|  | Std | 0.4 |
| a1 | Mean | 4 |
|  | Std | -1.496 |
| a2 | Mean | 0.5 |
|  | Std | 0.08 |
| a3 | Mean | 1.0954 |
|  | Std | -0.1496 |
| a4 | Mean | 0.3978 |
|  | Std | 0.3352 |
| b0 | Mean | 1.8169 |
|  | Std | 0.6667 |
| b1 | Mean | 1.7091 |
|  | Std | 0.1591 |
| b2 | Mean | 0.2484 |
|  | Std | 0.2163 |
| a0,a1 | Corr. Coef. | -0.3937 |
| **Noise magnitude mean curve coefficients (°C)** |  |  |
| a |  | -7.40E-12 |
| b |  | 4.22E-09 |
| c |  | -9.50E-07 |
| d |  | 1.12E-04 |
| e |  | -7.97E-03 |
| f |  | 3.38E-01 |
| **Noise magnitude std curve coefficients (°C)** |  |  |
| a |  | -1.00E-11 |
| b |  | 5.18E-09 |
| c |  | -1.01E-06 |
| d |  | 9.60E-05 |
| e |  | -5.13E-03 |
| f |  | 1.80E-01 |
| **Daily Maximum Temperatures (°C)** |  |  |
| a0 | Mean | 4.2024 |
|  | Std | 0.7175 |
| a1 | Mean | -1.4389 |
|  | Std | 1.6767 |
| a2 | Mean | 0.3221 |
|  | Std | 1.1248 |
| **Daily Minimum Temperatures (°C)** |  |  |
| a0 | Mean | 4.48 |
|  | Std | 0.7123 |
| a1 | Mean | -1.4289 |
|  | Std | 1.9805 |
| a2 | Mean | 0.1488 |
|  | Std | 0.1343 |
| **Radiation** |  |  |
| Solar constant (W·m-2) |  | 1395 |
| Autocorrelation of hourly "noise" |  | 0.78 |
| Latitude (radians) |  | 0.7854 |
| **Rainfall** |  |  |
| Rainfall rate (m·s-1) | Mean | 3.30E-07 |
|  | Std | 1.40E-07 |
| Probability of rain |  | 0.02 |
| Probability of rain if it is already raining |  | 0.8 |

* see Parrott et al. (1996) for a description of the temperature parameters.

**Table A.1** Configuration of the Weather Generator: List of parameter values used to generate the ecosystem forcing functions in all current simulations.

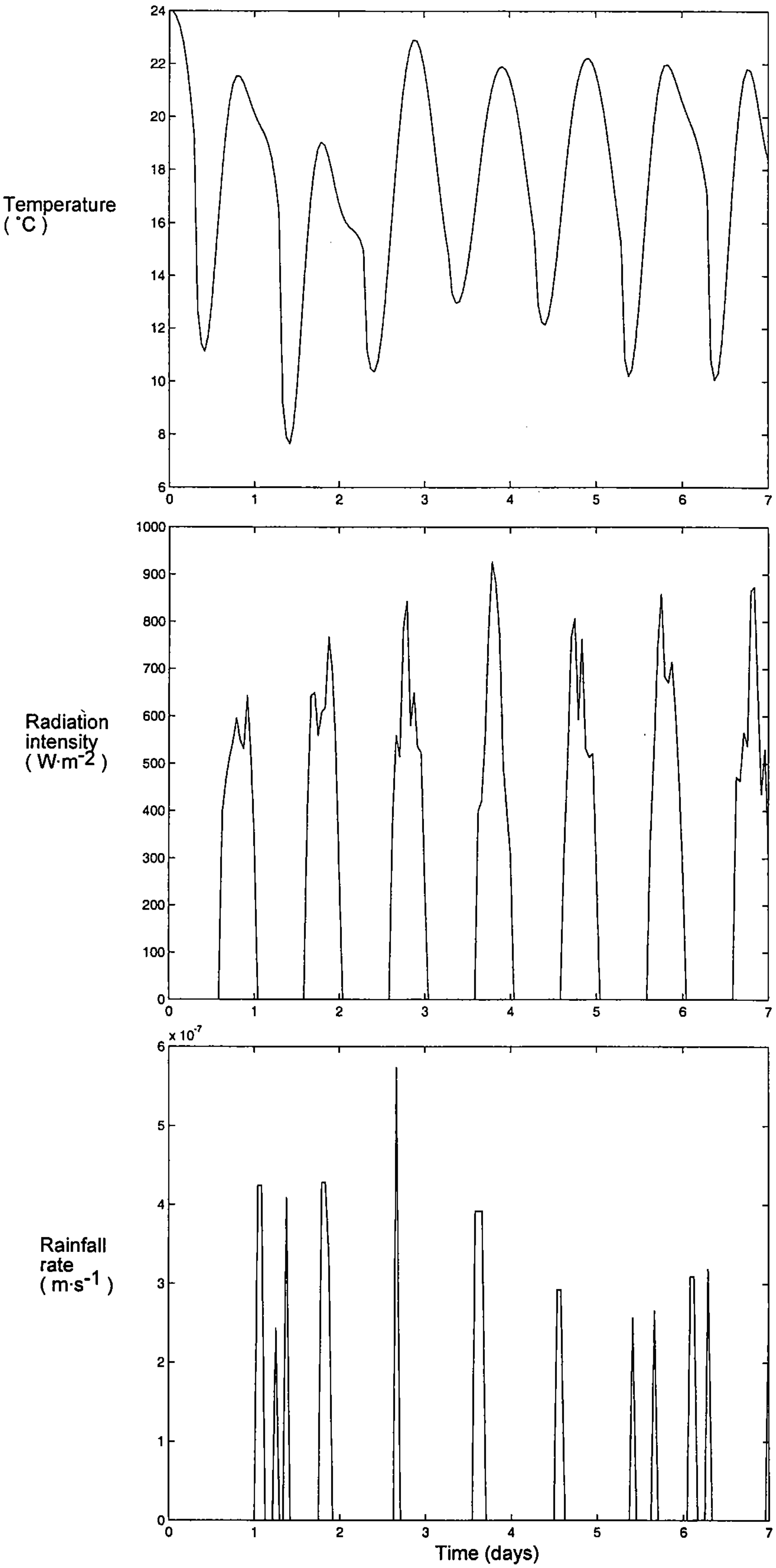


Figure A.1: One week of weather data created by the Weather Generator with the parameter values supplied for the ecosystem simulations.

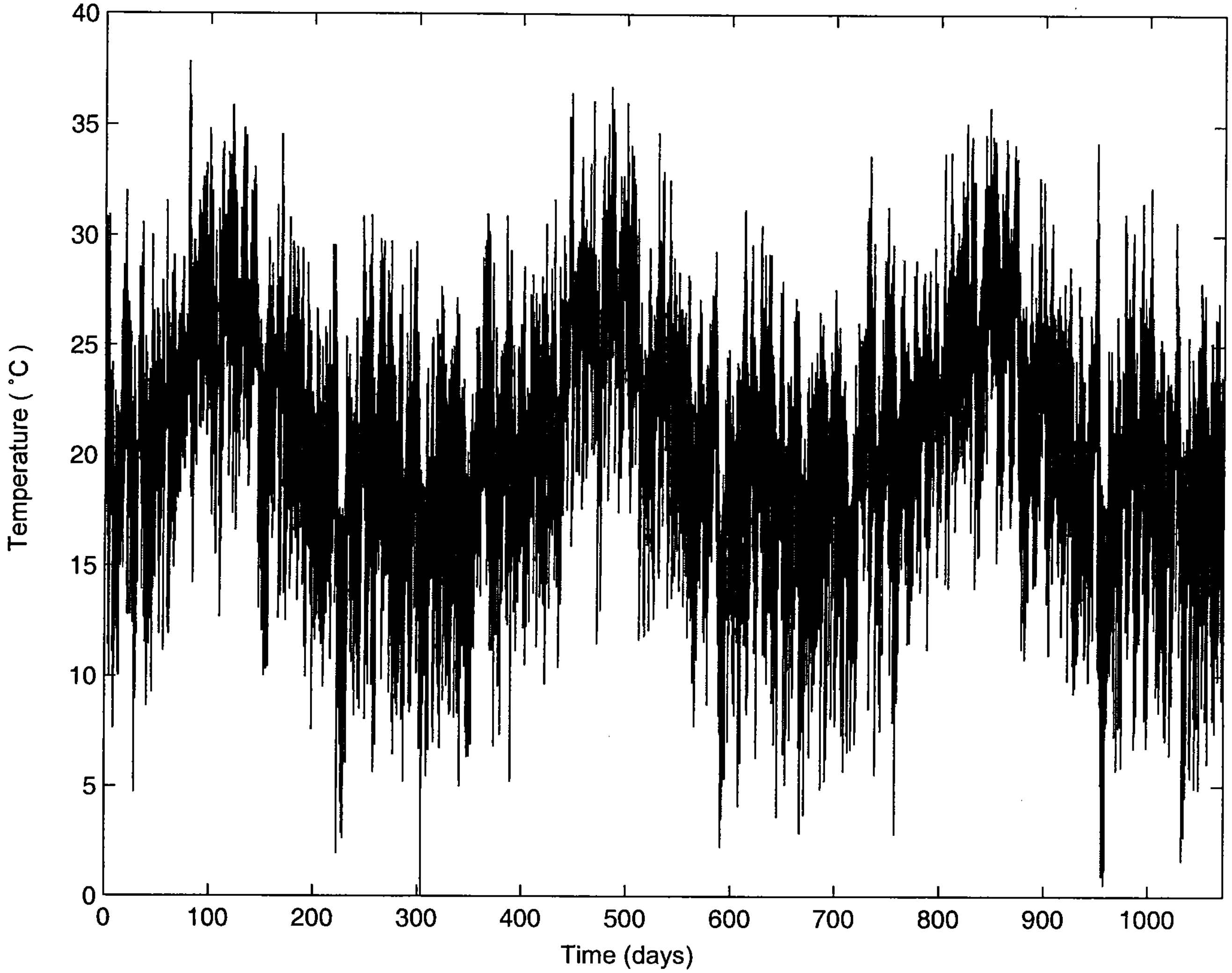


Figure A.2: Three years of temperature data created by the Weather Generator with the parameter values supplied for the ecosystem simulations.

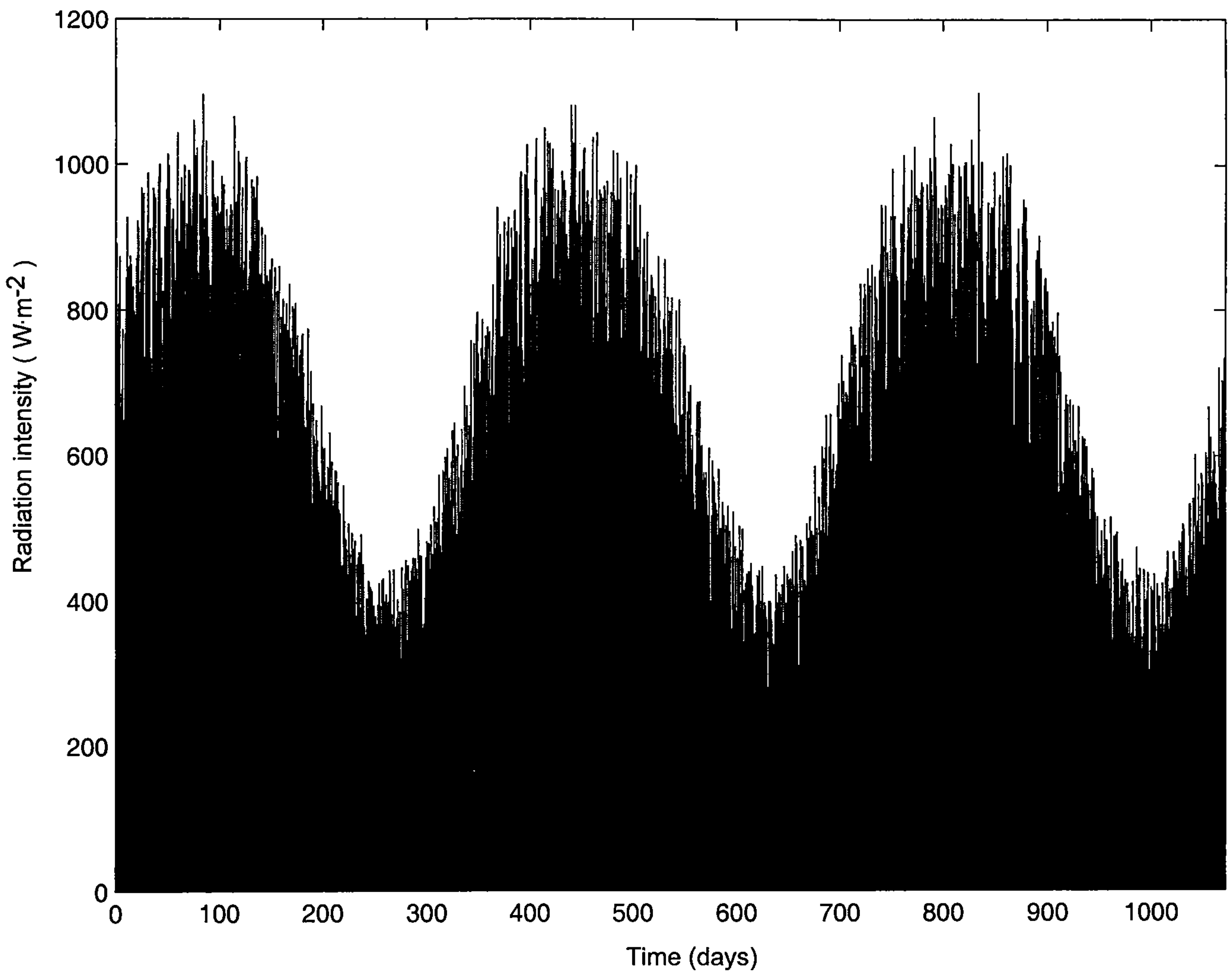


Figure A.3: Three years of radiation data created by the Weather Generator with the parameter values supplied for the ecosystem simulations.

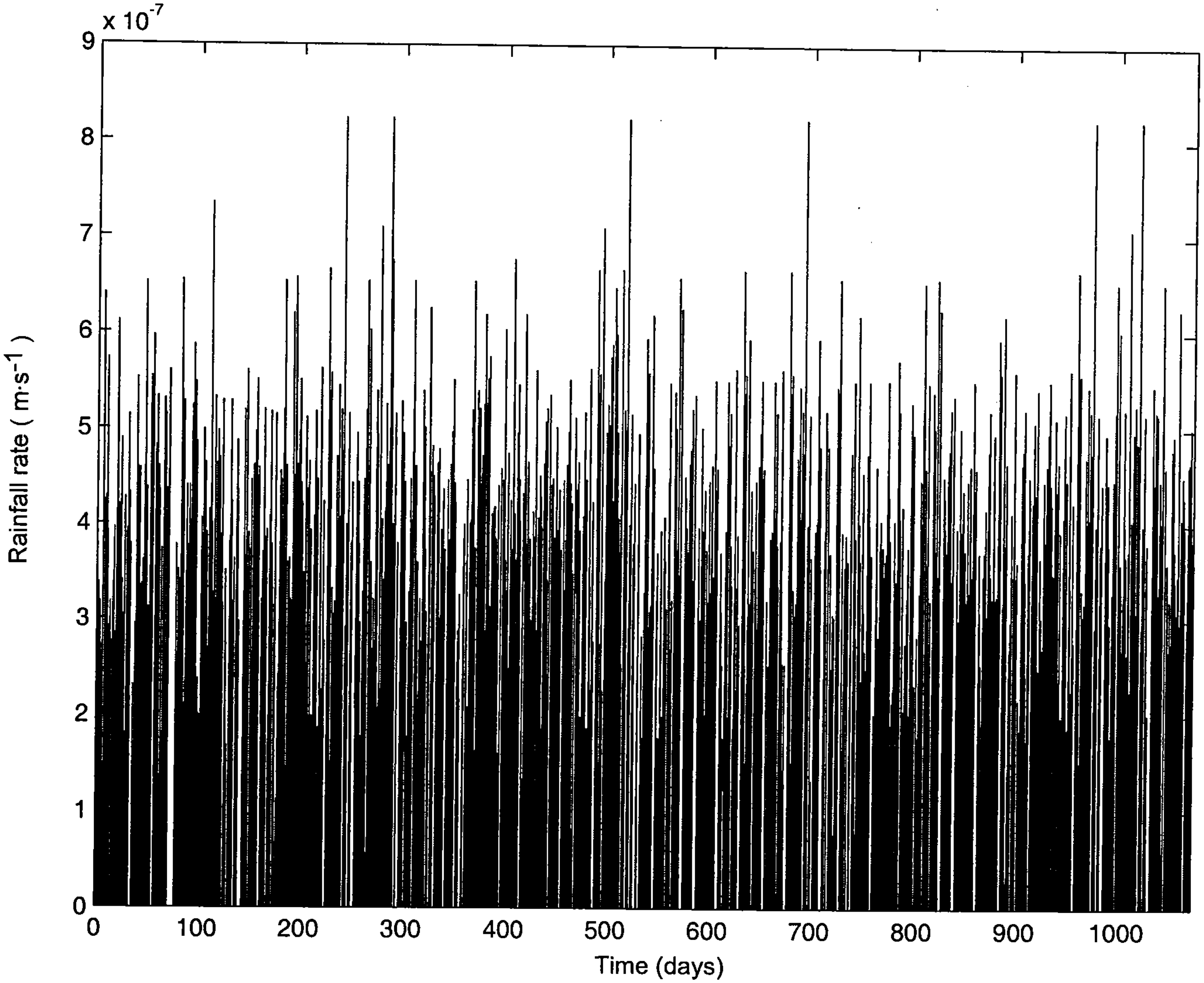


Figure A.4: Three years of rainfall rate data created by the Weather Generator with the parameter values supplied for the ecosystem simulations.

| **Yearly temperature** | | | |
|---|---|---|---|
| Average | 21.36°C | Std. | 0.05°C |
| Minimum | -4.47°C | Maximum | 40.05°C |
| **Yearly rainfall** | | | |
| Average | 0.8985 m | Std. | 0.0253 m |

Table A.2 50-year weather statistics for the climate forcing functions during ecosystem simulations.

### A.6. Discussion

The current set of weather routines meet the basic requirement of providing data that makes sense (i.e., the sun rises and sets at appropriate times, rainfall rates are moderate, and the temperature varies according to expected daily and annual cycles). In addition, the routines can generate these values as required by the simulation program for any time of the day or year. Lastly, the Weather Generator is completely configurable, so that different types of climates can be simulated and used as forcing functions to the ecosystem.

The Weather Generator could be improved in two major ways: first, the rainfall routines could be modified to incorporate the seasonal rainfall patterns that are evident in many climates, and second, the three streams of data ought to be correlated. These two adjustments will be implemented in future versions. As well, an alternative method of describing radiation patterns with a Fourier approach similar to that used for temperature data is being explored to improve the description of variations in radiant intensity due to atmospheric effects. Despite these few shortcomings, the performance of the current version of the Weather Generator is satisfactory and effectively serves the purpose of generating climate related forcing functions for the ecosystem model.

## A.7. References

Christie, B.R., ed. 1987. Handbook of Plant Science in Agriculture, Volume I. CRC Press: Boca Raton, Florida.

Parrott, L., R. Kok and R. Lacroix. 1996. Daily average temperatures: Modeling and generation with a Fourier transform approach. *Transactions of the ASAE* 39(5): 1911-1922.