



ELSEVIER

Ecological Modelling 93 (1996) 139–153

**ECOLOGICAL  
MODELLING**

# An object-oriented simulation framework for individual-based simulations (OSIRIS): *Daphnia* population dynamics as an example

Wolf M. Mooij<sup>\*</sup>, Maarten Boersma<sup>†</sup>

*Netherlands Institute of Ecology, Centre for Limnology, Rijksstraatweg 6, 3631 AC Nieuwersluis, The Netherlands*

Received 10 April 1995; accepted 4 December 1995

## Abstract

A general framework for the implementation of ecological models directed towards the falsification of knowledge, as opposed to models directed at making predictions, is proposed. The framework is constructed by defining a set of classes, with their interrelationships, in an object-oriented programming language. The classes represent the major levels of the so-called levels-of-integration hierarchy: *individual*, *population* and *system*. The abiotic physical and chemical environment is implemented by the classes *condition* and *resource*, respectively. Class *habitat* is used to represent the spatial structure of an ecosystem. The simulation is controlled by a class called *analyser*. The simulation mechanism is implemented by deriving all these real-life objects from a more abstract class *simobject*. The engine of the simulation is formed by a dynamic list of references to simobjects, sorted according to the time each simobject should be activated next. The data of each object are implemented in class *datobject*, from which simobject is derived. The applicability of this framework, called OSIRIS (object-oriented simulation framework for individual-based simulations), is shown for a population dynamical study on daphnids. The effects of variation among individual daphnids on the growth rate and structure of a population of daphnids are studied by comparing the results of the individual-based model with those of a life table. Moreover, variation in population growth rate over time, which parameter cannot be derived from a life table, is calculated. Finally, the sensitivity of the model for the number of modelled individuals and the sampling interval is analysed.

*Keywords:* Model; Falsification versus prediction; Levels of integration; Individuals versus ecosystem

## 1. Introduction

A variety of mathematical models is currently applied in ecological studies. Failure to recognise that different approaches underlay these models may

easily lead to the adoption of a type of model less suitable for the purpose intended (Hogeweg and Richter, 1982). Therefore, each modelling activity should start with a proper definition of the goal of the model (Starfield and Bleloch, 1986). Generally spoken, two goals can be distinguished: (1) falsification of knowledge and (2) making realistic predictions. To a certain extent these goals seem to be incompatible (Levins, 1966). These different goals have led to different types of models (Nisbet and Gurney, 1982).

<sup>\*</sup> Corresponding author. Tel.: +31-294-239300; fax: +31-294-232224; e-mail: mooij@cl.nioo.knaw.nl

<sup>†</sup> Present address: Max Planck Institute for Limnology, P.O. Box 165, 24302 Plön, Germany.

The goal of models directed towards the falsification of knowledge is 'to represent current knowledge as faithfully as possible so that this knowledge can be tested for sufficiency in generating the overall behaviour of the ecosystems, and thus reveal lacunae in our current knowledge' (Hogeweg and Richter, 1982). This implies that for these kind of models, negative results, i.e. discrepancies between model and behaviour and 'real world' observations, are in fact positive results, as they stimulate further research for new mechanisms. It is important to stress that even complete knowledge of the causal relationships within a system does not necessarily lead to predictive power as is shown by simple deterministic models that show chaotic behaviour for part of their parameter range (May, 1975; Scheffer, 1990). In their most abstract form, knowledge-oriented models are sometimes referred to as mini models (Hogeweg and Richter, 1982). Well known examples of such mini models are the Lotka–Volterra predation and competition models (Yodzis, 1989). More complex knowledge-oriented models share their expanding number of parameters and dependency on data with prediction-oriented models. Because falsification of hypothesis is the essential objective of knowledge-oriented models, such models should in principle only included those state variables and parameters that are measurable under experimental conditions.

Prediction-oriented models differ from knowledge-oriented models in their objective. Their goal is to produce realistic and reliable output, mostly for management purposes, and as a consequence of this they depend heavily on field data. In their extreme, these models may consist of neural networks which relate the output with the input using a black box of hidden layers (Sarle, 1994). Mostly however, a more reductionistic approach is followed. The ecological system under study is divided in compartments and the fluxes between these compartments are expressed in a set of differential equations. Together, these equations form a simulation model. This approach results in large numbers of state variables, process formulations and parameters. The value of the parameters is either determined a priori by expert judgement (Jørgensen et al., 1991) or a posteriori by calibration procedures. The suggestion that these models properly represent the causal relationships of the studied system is under criticism (Peters, 1990;

Van Tongeren, 1995). Indeed, Klepper (1989) showed that a wide range of parameter sets yielded the desired relation between input and output in a 80 parameter compartment model of an aquatic system. This shows that whereas the model as a whole may produce reliable output (the prediction-oriented goal), no conclusions can be drawn about the validity of its components (the knowledge-oriented goal).

Because of their applied nature, a wide variety of prediction-oriented models is currently used by ecologists. When prediction is the main target, it pays to be opportunistic. For knowledge-oriented models, however, which strive for a mechanistic explanation of the studied phenomena, we may expect some degree of generality in the way ecological systems are represented. This generality expresses our current conceptualisation of ecological systems, as described in basic textbooks (i.e. Begon et al., 1990). Some attempts have already been made to create a more formal description of this conceptual framework (Baveco and Lingeman, 1992; Maley and Caswell, 1993). Such a framework can serve as a template for a knowledge-oriented model of a specific ecological system. We want to contribute to this process of defining a basic format for a knowledge-oriented model of an ecological system.

To reach this goal we start with a survey of the entities, with their interrelationships, that should form the building blocks of such a framework. We then set out to formalize these entities by implementing them as classes of an object-oriented programming language. At first hand, the translation of a concept into source code seems to be superficial. However, it is important to recognize that there is often a tight relation between the development of a concept and the tools to represent it. For instance, the development of mathematics would not have been possible without a mathematical notation. The suitability of object-oriented programming languages for the implementation of individual-based models has already been stressed by Baveco and Lingeman (1992) and Maley and Caswell (1993). We have come to the conclusion that there are three levels of information involved which can be represented by means of the inheritance mechanism of object-oriented programming (Meijer, 1988). We called the resulting framework OSIRIS (object-oriented simulation framework for individual-based simulations), after the egyptian

god whose body, after being cut to pieces, was restored by putting all the parts together. To show the applicability of OSIRIS, we describe a *Daphnia* application which we have built within the framework. We have fed the model with experimentally measured life-history parameters of individual *Daphnia* and studied the resulting output parameters from a population dynamical point of view.

## 2. Entities within ecological models

Individuals seem to be the most self-evident units to be represented in knowledge-oriented models (Hogeweg and Richter, 1982). At least in animal ecology, they are the best defined entities (MacMahon et al., 1981). An increasing number of papers has appeared following this individual-based approach (Huston et al., 1988; DeAngelis and Gross, 1991; Van Winkle et al., 1993). Individual-based models allow to take two basic aspects of ecological systems into account which were paid little attention to in traditional modelling. First, individuals vary in their phenotype from each other (Chambers, 1993; Tyler and Rose, 1994) and secondly, interactions between them take place on a local scale (Mccauley et al., 1993; Wilson et al., 1993).

However, an ecological model cannot be built from individuals only. This holds of course for the modelling of the abiotic physical and chemical environment, but also for some of the biotic components. Starfield and Bleloch (1986) stress that to be successful in the modelling of ecological systems the scope and detail of the model should be chosen carefully: the choice for modelling individuals in the focus of the model does not imply that biological components on the border of the model should also be represented at this level of detail. For instance, the food of an organism which is modelled at the individual level may be represented at the population level or even as a functional group.

The insight that ecological processes can be observed with varying scope and detail is expressed in the hierarchy of 'levels of integration' (Begon et al., 1990). Within an ecological context these levels may range from single genes to whole landscapes. A more common, restricted list contains the following levels: individual, size group, population, functional

Table 1

A hierarchical representation of the levels of integration which are currently used by ecologists. From this general list, two selections of entities can be made, one used in production studies and one used in population dynamics

Levels of integration	Production studies	Population dynamics
Individual		Individual
Size group	Size group	↓
Population	↑	Population
Functional group	Functional group	↓
Community	↑	Community
Ecosystem	Ecosystem	

group, community, ecosystem (Table 1). We agree with Polishchuk (1992) that this list, although self-evident at first sight, is in fact a combination of two lists, originating from different viewpoints, one production-oriented and the other focusing on population dynamics. A major difference between production-oriented and population dynamical studies is the unit by which biological tissue is described, biomass respectively individuals.

Production-oriented compartment models of a certain area have a long tradition and formed in the late sixties the basic concept behind the International Biological Programme (Cameron and Billingsley, 1975). These type of models are often prediction-oriented. Within this formalism, an ecosystem is a spatial unit with relative little exchange of nutrients with the surrounding world. The nutrient dynamics within a certain ecosystem are modelled by grouping organisms according to similarities in their food resources and their predators in functional groups (DeAngelis, 1992; Pauly and Christensen, 1995). However, because many species have different food types and/or predators during their ontogeny, different size classes of a single population may be found in different functional groups. The basic criticism from a knowledge-oriented point of view on the concept of functional groups is that the criterium by which organisms are pooled in poorly defined groups. Therefore, once measured, the parameters of a functional group have a limited relevance for other ecosystems and it is difficult to design experiments in which they can be measured accurately.

The population dynamical hierarchy of an individual, which belongs to a population, which is part of a

community, seems a better starting point for a knowledge-oriented model. At least in animal ecology, individuals are clearly defined and recognizable. Given a certain geographical area and a clear taxonomy, also a population is, at least theoretically, a properly defined entity. A community, however, is a more flexible entity (Diamond and Case, 1986). We agree with Begon et al. (1990) that the ecosystem and the community represent the same level of biological complexity. They see an ecosystem as a combination of the whole biotic community and the abiotic physical and chemical environment.

The abiotic components of an ecosystem usually taken into account are environmental parameters such as temperature, light, nutrients and many others. Begon et al. (1990) split these parameters into conditions and resources. In their definition, conditions are those environmental factors that affect components of the ecosystem, but are not consumed (e.g. temperature). Abiotic resources are also external factors but because they are depleted, there may be competition for them (e.g. nutrients).

From this survey the following picture emerges. An ecosystem is a concept which integrates the concept of spatial structure with the concept of a resource structure. The spatial units, or habitats, are linked to form a spatial network, whereas the resources are linked to form a trophic network. By definition, resources interact on a local scale, that is within habitats. We will further refer to these local resources as 'resource items'. Resources *sensu lato* can take many forms. We suggest to distinguish three types of resources: conditions, resources *sensu stricto* and populations, following the major disciplines of the natural sciences: physics, chemistry and biology. In our definition, condition items are those entities which dynamics fall within the scope of physics, such as light or temperature. Resource items are those entities which, besides being governed by the law of physics, also show chemical activity, such as nutrients and detritus. The living objects, which show besides physical and chemical activity also biological activity, are represented as individuals, grouped in populations.

We conclude that a knowledge-oriented model of an ecological system can be built on the basis of the following basic concepts: ecosystem, habitat, condition, condition item, resource, resource item, popula-

tion and individual. Using this approach, the state of a system at a given moment in time can be given in a relational database which lists all the individuals, habitats, etc. To simulate the system, physical, chemical and biological process formulations have to be added to modify the records of these databases over time. The link between a record and a set of functions that acts on its contents is the basic concept behind object-oriented programming. In the jargon of this programming concept, the description of the format of the record and the algorithms of functions that act on its contents is called a class. The actual records, which contain the data, are called objects. At the creation of a new object, the class definition acts as a template.

### 3. OSIRIS framework

In the tradition of program design by functional decomposition (Yourdon, 1975) the development of a program starts with definition of the context in which the programme should function. In a so called context diagram the relations between the programme, which is still seen as a black box, and the outside world are listed (Fig. 1). For a simulation programme typically three kinds of information have to be passed to and received from the model. (1) Before starting each simulation the model has to be fed with the databases which describes the state of the system at the beginning of the simulation (*inp files*), whereas the model returns the databases that describes the state of the system at the end of each simulation (*out files*). (2) To keep track of the state of the system during simulation, the values of state variables and process rates can be saved with a given time interval (*log files*). (3) To document the different runs of the model, all the above mentioned files are listed in a set up file (*ini file*). Error messages produced by the programme are written to an error file (*err file*).

As a next step OSIRIS is decomposed. The functional decomposition of a system for implementation in an object-oriented programming language such as C++ (Stroustrup, 1991) or SMALLTALK (Pinson and Wiener, 1988) differs from that of a procedural language such as FORTRAN or Pascal. In the object-oriented programming concept two types of rela-

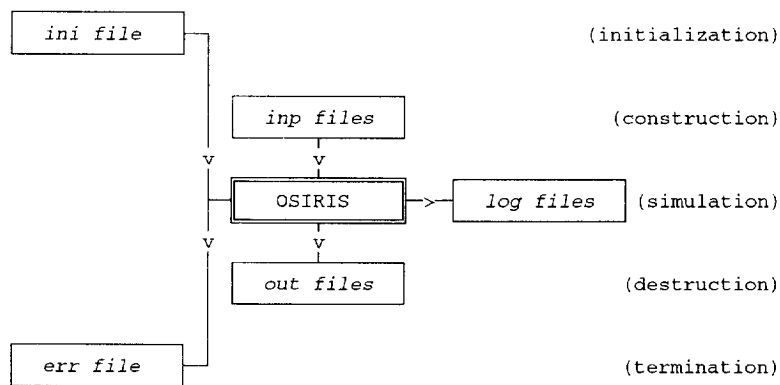


Fig. 1. Context diagram of OSIRIS showing the five major steps in the simulation process (initialization, construction, simulation, destruction and termination) and the files that are involved in each step.

tions among components of the system exist. These relations are called HASA and ISA relations (Meijer, 1988). In the procedural programming concept language only HASA relations are found.

ISA (from 'is a') relations express levels of abstraction. For instance, a human is also a primate, a mammal, a vertebrate, an organism, a body of mass, an entity (and not vice versa). One of the essential innovations of object-oriented programming is that these levels of abstraction, which play an important role in human language, can be expressed directly by means of a mechanism called inheritance. By referring in the class definition to another class as 'base class', the variables and the functions of that base class are automatically inherited. Classes that refer to base classes are called 'derived classes'. The challenge of object-oriented programming is to imple-

ment variables and functions at the highest possible level of abstraction. For instance, the number of vertebrae is a variable shared by all vertebrates and so it would be unwise to implement it at the level of a primate or a human. By doing so, all classes which are derived from vertebrae can benefit from this variable and the duplication of code is avoided.

In OSIRIS we chose to distinguish four levels of abstraction (Fig. 2). At the most specific level (4) the ecological recognizable units are implemented as the classes System, Habitat, Condition, ConditionItem, Resource, ResourceItem, Population and Individual. To show the overall picture, all these classes are referred to as a single class BioObject in Fig. 2. Their interrelationships, which are discussed in depth in the following paragraphs, are implemented by means of pointers, arrays and Btrees and two addi-

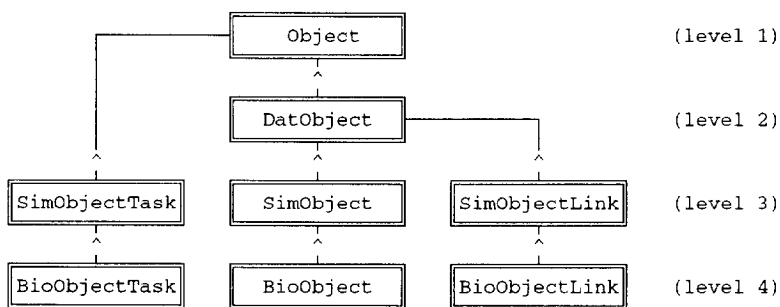


Fig. 2. ISA relations within OSIRIS (BioObject: one of Analyser, System, Habitat, Condition, ConditionItem, Resource, ResourceItem, Population or Individual).

tional classes, HabitatLink and ResourceLink. Both link classes are referred to as class BioObjectLink in Fig. 2. Although different in their biological meaning, these classes share that we want to simulate their behaviour. Therefore, they are all derived from a class called SimObject. At the SimObject level of abstraction (3), the simulation mechanism is implemented, with the help of the class SimObjectTask. At the last but one level (2) only the static properties of an object are left in a class called DatObject. At this level an object is seen as a record in a database, with a given number of integer, real and string fields. DatObject is derived from Object, a class without any data or functionality, which forms the basic level of abstraction (1). The derivation of every class from one abstract class, which is an essential concept in SMALLTALK (Pinson and Wiener, 1988), enables us to benefit from a so called object-based library of container classes (Borland International, 1993). The benefit of such libraries is that complex data structures such as expandable arrays and balanced Btrees (Knuth, 1969) are readily available.

HASA (from 'has a') relations are the type of relations that form the structure of a relational database. In a database system which is saved on disk, HASA relations are implemented by including the same field in the records of two different databases. However, when loaded in the memory of a computer, a more efficient way to implement HASA relations between objects is by means of pointers. In the case of OSIRIS, these pointers represent relations such as those between two spatially linked habitats or between an individual and its population. When the relation points to more than one object, an array or a Btree of pointers is applied (Maley and Caswell, 1993).

The implementation of the HASA relations among the biological classes is straightforward (Fig. 3a and b). Individuals have a pointer to their population and a pointer to the habitat in which they currently live. Populations keep track of their individuals by means of a Btree. In the System object, the Populations, Habitats, Conditions and Resources that are included are listed in arrays. To enable Habitats, Conditions, Resources and Populations to communicate with each other, each of these objects has a pointer to the System object. The spatial structure of the modelled system is implemented by HabitatLink objects. These

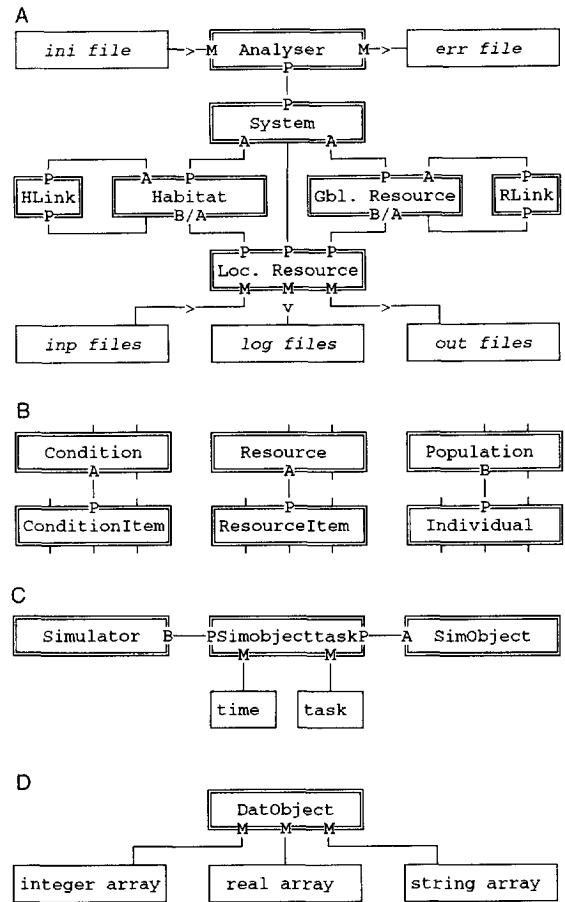


Fig. 3. HASA relations at the BioObject level (M = member, P = pointer, A = array, B = Btree). For reasons of simplicity, the classes Condition, Resource and Population are shown as one class GlobalResource, whereas the classes ConditionItem, ResourceItem and Individual are depicted as the class LocalResource. (b) HASA relations at the BioObject level (P = pointer, A = array, B = Btree), showing the three pairs of global and local resources in OSIRIS, which were combined in (a). (c) HASA relations at the SimObject level (M = member, P = pointer, A = array, B = Btree). (d) HASA relations at the DatObject level (M = member).

objects contain pointers to the Habitats that they connect. The trophic structure of the system is implemented by ResourceLink objects. These objects are not modelled themselves (Fig. 2) but contain data about the link they represent. During construction of the system, the number of objects of each class and their initial values are read from *inp files*. When the system is destructed these data are written to *out files*. During simulation objects of each class may

write information to *log files*. There is one *inp file*, *out file* and *log file* per class. In Fig. 3a these files are only shown for a local resource class. The class Analyser (Fig. 3a) is used to study and manage the behaviour of the system in a series of computer experiments. For example, it would be possible to implement formal numerical analysis algorithms (Klepper and Rouse, 1991) as a function of this class in order to study the behaviour and sensitivity of the model. Because the Analyser controls the whole simulation, the *ini file* and *err file* are connected to this object.

Each of the above mentioned biological classes shares the property that we may want to perform some task in the simulation process. This functionality is implemented with the help of three classes: SimObject, SimObjectTask and Simulator (Fig. 3c). The Simulator object, of which there is only one, activates the SimObjects that we want to simulate one by one. All SimObjects that are active are listed in a Btree. The SimObjects are sorted according to the moment that they should be activated. After being simulated, a SimObject returns the time it should be activated again to the Simulator, and is scheduled accordingly. However, SimObjects are not listed themselves in the Btree of the Simulator. This is because we may want to have one SimObject to perform several tasks. Therefore, an intermediate data structure, SimObjectTask, is created. Each SimObjectTask is linked to a specific SimObject by means of a pointer. Moreover it contains a pointer to the function of its SimObject that it should perform when activated and it contains the time at which this should happen. Each SimObject has an array of pointers to its tasks, whereas the Simulator has a Btree of pointers to all tasks.

Data are implemented in the class DatObject (Fig. 3d). Each DatObject has three arrays connected to it, which contain the data as integers, reals and strings. The main functions of DatObject is to read the data from disk during construction of the system, to control access to them during simulation and to write them to disk during destruction of the system.

Although the object-oriented programming concept is essential to our approach the actual language in which the source code is written is a matter of taste. We chose C++ for a number of reasons. One is the availability of standard C++ compilers for a

wide number of platforms (PC, workstation, main-frame) and operating systems (DOS, Windows, OS/2, Unix) which guarantees an optimal portability. Moreover, extensive mathematical libraries are available for C/C++ (Press et al., 1992; Dyad Software Corporation, 1992). Finally, we expect a superior run-time performance of C++ compared to other object-oriented languages such as SMALLTALK.

#### 4. Daphnia application

OSIRIS was designed to host a wide variety of applications. Because we wanted to focus this paper on the framework but not so without showing a possible application, we looked for a relatively simple problem, which is still complex enough to be interesting. Generally speaking, complexity can be entered in an application in two ways: by creating a complex spatial structure and or by creating a complex resource structure. Cellular automata such as Life (Sigmund, 1993) are an example of the first type, whereas individual-based models (DeAngelis and Gross, 1991) are of the second type. As freshwater ecologists (Mooij and Van Tongeren, 1990; Mooij et al., 1994; Boersma and Vijverberg, 1994a, Boersma and Vijverberg, 1994b), we chose to use an individual-based model of the population dynamics of *Daphnia galeata* as an example. In this real life application, we used the capability of OSIRIS to easily create, monitor and destruct large numbers of individuals to study the consequences of phenotypic variation on the growth rate and structure of the whole population. As a result, the classes Individual and Population play a central role in this application. Because we took no relations to food or abiotic factors into account and did not look for spatial processes, the classes Condition, Resource and Habitat are only present in a rudimental form without any data or functionality. The class System is used to rebuild the system for several parameter settings, under control of class Analyser.

Data on individual daphnids were obtained from Boersma and Vijverberg (1994a). They measured life-history traits of individual *Daphnia galeata* in the laboratory at a fixed temperature (17.5°C) and food supply (2.5 mg C l<sup>-1</sup>) well above the incipient

limiting level. From these data on the duration and specific fecundities of the moulting stages — further referred to as instars — the intrinsic rate of population growth ( $r$ ) was calculated by means of a life table (Stearns, 1992). However, because a life table only takes the average parameters of each instar into account, this approach does not allow to evaluate consequences of the experimentally measured variance in these parameters on the growth rate and structure of the population. Including this variation may lead to a different estimate of the average population growth rate or structure, compared with the results of a life table. For certain, variability among individuals in instar duration and fecundity will lead to a variation in the population growth rate when measured repeatedly with a certain time interval. This variation in  $r$  will depend on the number of individuals and the interval at which the population growth rate is measured. The expected relation between the measured variations  $s_r(n, \Delta t)$  for a certain number of individuals ( $n$ ) and a given time step ( $\Delta t$ ) with variation in  $r$  of one individual and one unit of time ( $s_r$ ) is (Sokal and Rohlf, 1982):

$$s_r(n, \Delta t) = s_r / \sqrt{n * \Delta t}. \quad (1)$$

The description and functionality of an individual *Daphnia* in our application is a simplification of the INSTAR model (Hogeweg and Richter, 1982; Vijverberg and Richter, 1982). In this discrete event model, the life of a *Daphnia* is simulated each time a new instar is reached. Each individual is characterized by two state variables: the current instar number ( $I$ ) and, if mature, the current number of eggs ( $F$ ). To simulate a *Daphnia* in this formalism we need four functions, describing maturation, juvenile and adult instar duration and fecundity (Table 2). Rather than choosing functions on theoretical grounds we adopted a more descriptive approach, because this enabled us to minimize the number of parameters of the model. Of the 37 individual *Daphnia* for which data were available only one of them matured after 5 instars, the others after 4 instars. Because we were not able to make a reasonable estimate of the variance in maturation from this data we ignored it in the current model and chose a fixed number of four juvenile instars. All three other functions could be described by linear relationships, except for the first clutch size, where we needed an extra parameter

Table 2

The four process formulations of an individual in the *Daphnia* application, describing maturation, juvenile and adult moulting stage (instar) duration and fecundity, respectively

Number of juvenile instars (JI)	
$JI = \alpha_{JI}$	(1)
Juvenile instar (JI) duration (JD):	
$JD = (\alpha_{JD} + \beta_{JD} * JI) * e^{\text{normal}(0, \gamma_{JD})}$	(2)
Adult instar (AI) duration (AD):	
$AD = (\alpha_{AD} + \beta_{AD} * AI) * e^{\text{normal}(0, \gamma_{AD})}$	(3)
Adult instar (AI) fecundity ( $F$ ):	
$F = \delta_F * (\alpha_F + \beta_F * AI) * e^{\text{normal}(0, \gamma_F)}$	(4)

(Figs. 4 and 5). To model the measured variance in both traits, we multiplied the resulting instar specific duration or fecundity with a lognormal distributed deviate, resulting in a proportional rather than a constant variance. Together, these four functions have 11 parameters (Table 3). These parameters we estimated from the experimental data on individual *Daphnia* by means of the maximum-likelihood technique.

The list of state variables, functions and parameters contains all the information necessary for the simulation of a given initial *Daphnia* population. Both the life-table and the individual-based model result in an exponential growing *Daphnia* population and therefore produced comparable estimates of the population growth rate. For the individual-based

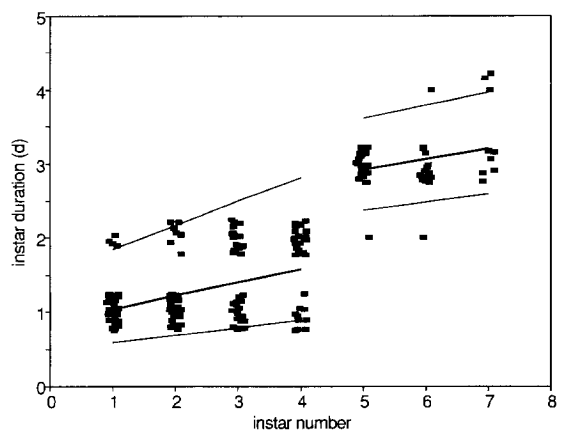


Fig. 4. Juvenile (1–4) and adult instar (4–7) duration of *Daphnia galeata*. The lines indicate the 95% prediction limits of the functions applied in the model. Data points are scattered both horizontally and vertically to show more of them.



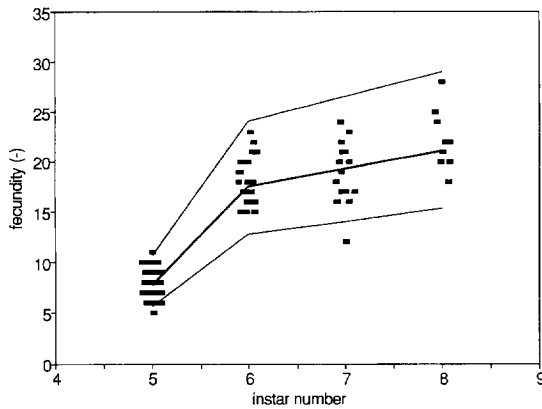


Fig. 5. Fecundity of the first four adult instars (5–8) of *Daphnia galeata*. The lines indicate the 95% prediction limits of the functions applied in the model. Data points are scattered horizontally to show more of them.

model the population growth rate  $r$  after a certain time step  $\Delta t$  can be calculated from the formula

$$r = (\ln(n_t) - \ln(n_0)) / \Delta t, \quad (2)$$

in which  $n_0$  and  $n_t$  represent the number of *Daphnia* at time 0 and  $t$ . Different strategies can be chosen to avoid the problem of exhaustion of computer memory due to the exponential increase of the number of *Daphnia*. One is to remove  $(n_t - n_0)$  randomly chosen animals after  $\Delta t$ , thereby resetting the number of animals to  $n_0$ .  $\Delta t$  has then to be chosen small enough so that the available memory is not exhausted. At its extreme, we can decrease  $\Delta t$  until  $(n_t - n_0) = 1$ , and hence we remove a ran-

Table 4

Population-dynamical output OSIRIS ( $n = 512$ ,  $\Delta t = 64$ ,  $r = 0.345$ ,  $s_r = 0.799$ ), showing the instar specific average instar duration (d. Durat), fecundity (Fecun), age (d. Age), relative survival (Surv), relative abundance (Abun) and reproduction (Repro). For a full description of each variable see Stearns (1992)

Instar	Durat	Fecun	Age	Surv	Abun	Repro
1	1.08	0	1.08	0.689	0.309	0
2	1.27	0	2.32	0.452	0.237	0
3	1.46	0	3.73	0.276	0.175	0
4	1.65	0	5.32	0.159	0.121	0
5	2.94	7.19	8.18	0.058	0.102	0.458
6	3.09	17.83	11.23	0.020	0.037	0.350
7	3.23	19.62	14.42	0.007	0.013	0.128
8	3.38	21.37	17.75	0.002	0.005	0.044
9	3.53	23.10	21.22	0.001	0.001	0.014
10	3.69	24.76	24.83	0.000	0.000	0.004
11	3.83	26.78	28.50	0.000	0.000	0.001
12	3.99	28.86	33.89	0.000	0.000	0.001

domly chosen individual immediately after a new individual is born. To keep track of the number of animals born over a period  $\Delta t$ , a separate counter  $\Delta n$  is increased each time an animal is born. Because the number of individuals is now constant, the population growth rate over a period  $\Delta t$  should now be calculated as:

$$r = \Delta n / \Delta t. \quad (3)$$

When sampled repeatedly, the standard deviation in  $r$  for a given number of modelled individuals  $n$  and a certain time step  $\Delta t$  can be calculated. To compare this standard deviation for different combinations of  $n$  and  $\Delta t$ , it should be normalized with Eq. 1.

To compare the structure of the population in the individual-based model with the structure that follows from a life table we sampled each of the relevant population dynamical parameters in the model. These parameters are instar duration, age, relative survival, relative abundance, fecundity and reproduction (Stearns, 1992). For each instar the arithmetic mean of each parameter was calculated (Table 4). To make the comparison with the life table as fair as possible, we used the instar duration and fecundity from the individual-based model as input. From these two parameters two other columns of the life table (Table 5) can be generated and  $r$  can be calculated using the Euler equation (Stearns, 1992). Although there are small differences between

Table 3

The eleven parameters of the four process formulations of an individual in the *Daphnia* application. The values of the parameters were estimated from individual cultures by means of the maximum-likelihood technique

$\alpha_{J1}$	number juvenile instars (-)	4.0000	(1)
$\alpha_{J2}$	intercept juvenile instar duration (d)	0.8577	(2)
$\beta_{J2}$	slope juvenile instar duration (d)	0.1800	(3)
$\gamma_{J2}$	CV juvenile instar duration (-)	0.2936	(4)
$\alpha_{A1}$	intercept adult instar duration (d)	2.7817	(5)
$\beta_{A1}$	slope adult instar duration (d)	0.1433	(6)
$\gamma_{A1}$	CV adult instar duration (-)	0.1089	(7)
$\alpha_F$	intercept adult instar fecundity (-)	14.0785	(8)
$\beta_F$	slope adult instar fecundity (-)	1.7566	(9)
$\gamma_F$	CV adult instar fecundity (-)	0.1623	(10)
$\delta_F$	reduction fecundity first adult instar (-)	0.4934	(11)



Table 7

Percentile deviations of CV<sub>r</sub> for different combinations of  $n$  and  $\Delta t$ , relative to the grand average of CV<sub>r</sub> for runs with  $t \geq 32$  (shown in italics)

$\Delta t$	$n$													
	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192
1/8	68	79	104	133	153	169	178	184	188	187	173	194	180	177
1/4	58	82	106	129	150	167	173	178	186	184	183	183	189	228
1/2	57	82	105	124	145	160	169	171	176	170	171	179	140	162
1	55	76	99	117	135	146	150	159	169	127	162	136	151	193
2	47	70	88	101	109	117	109	121	124	145	125	105	103	124
4	33	51	66	74	71	71	62	72	94	74	63	92	74	62
8	0	10	25	27	40	33	35	25	48	40	35	36	33	
16	-4	7	15	23	15	27	22	18	10	14	20	6		
32	-8	1	<i>15</i>	8	<i>11</i>	<i>11</i>	-5	-8	2	-6	-1			
64	-1	-9	-3	<i>15</i>	<i>10</i>	<i>12</i>	-8	2	-6	<b>6</b>				
128	-9	2	6	3	17	-4	-7	-7	-6					
256	-3	-8	5	5	-1	-3	4	-10						
512	-5	-6	3	7	9	1	-9							
1024	-9	-7	-7	<i>13</i>	1	-12								

*Daphnia* populations in nature will normally be many orders of magnitude larger than the largest modelled population, deviations of runs with a smaller  $n$  from the runs with the higher value of  $n$  should be seen as artefacts. A visual inspection of Figs. 8–10 shows that below 128 modelled individuals the relative survival, relative abundance and relative reproduction of the different instar begin to shift. This is much earlier than the shift in the population growth rate which becomes only pronounced below 16 individuals (Fig. 6).

Concluding, we can say that for  $n \geq 512$  and  $\Delta t \geq 32$ , the *Daphnia* application gave stable results in terms of  $r$ ,  $s_r$  and the instar specific survival, abundance and reproductive output. Moreover, these runs were feasible in terms of memory and computational power requirements. What is won compared to a life table is evidence that effect of the variation in life-history traits on  $r$  is relatively small. Moreover, we now have estimates of the standard-deviation in population growth rate, and standard deviations of

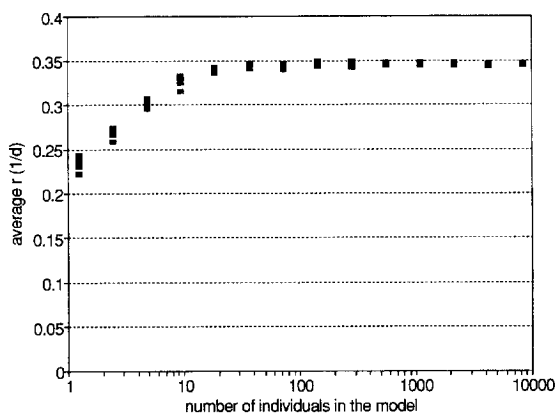


Fig. 6. Average population growth rate as function of the number of modelled individuals.

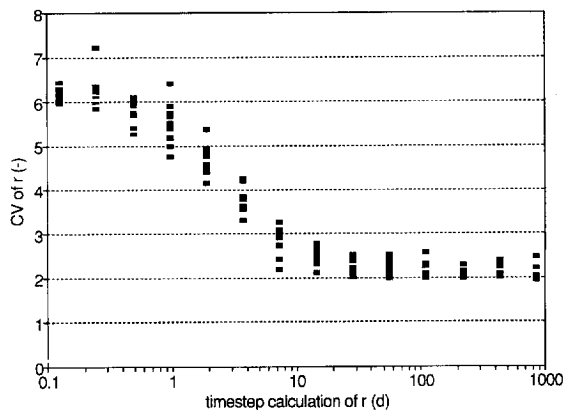


Fig. 7. Coefficient of variation of the population growth rate as a function of the time interval between measurements.

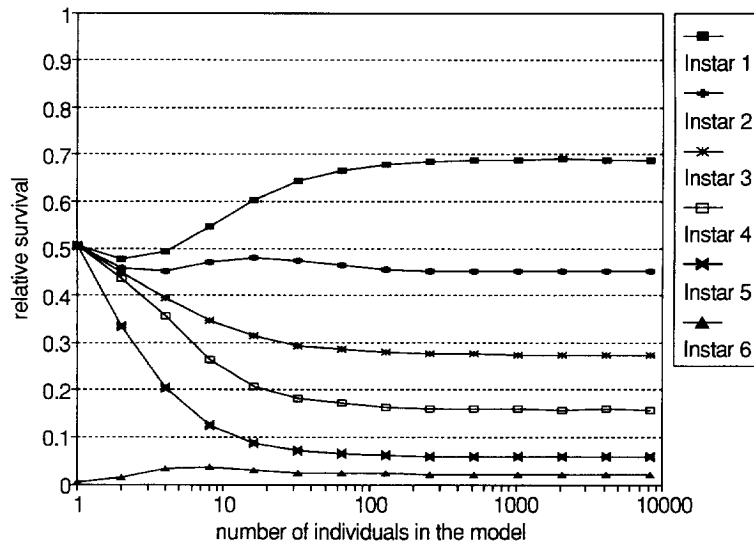


Fig. 8. Instar specific relative survival as a function of the number of modelled individuals.

the instar specific relative survival, relative abundance and reproduction. It will be possible to falsify these results in future experiments. *Daphnia* populations can be reared in the laboratory and sampled at a time interval  $\Delta t$ . From a series of these samples  $r$

and  $s_r$  can be calculated and compared with the model results. By doing so, we would test the hypothesis that the population dynamics of *Daphnia* under a fixed set of environmental conditions can be explained from the mere interaction of the life-his-

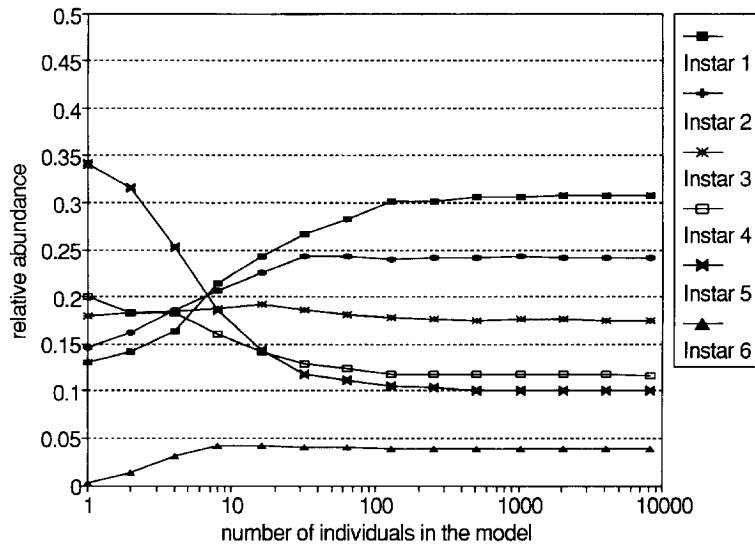


Fig. 9. Instar specific relative abundance as a function of the number of modelled individuals.

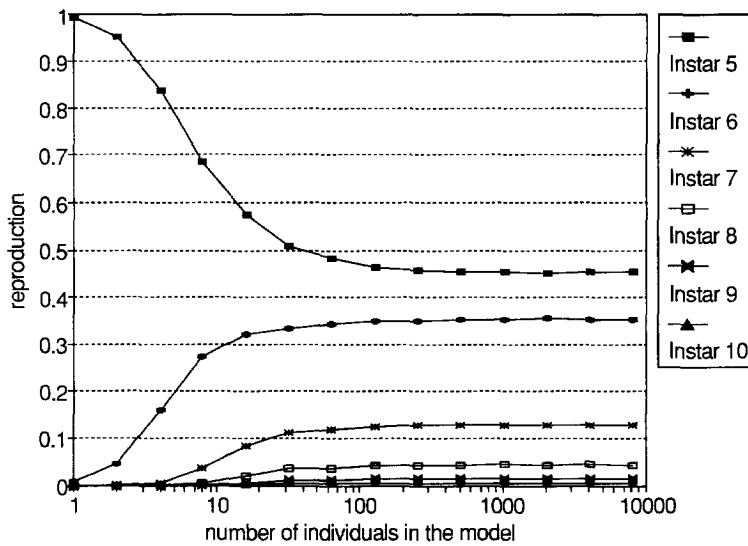


Fig. 10. Instar specific relative reproduction as a function of the number of modelled individuals.

tory traits instar duration, instar at maturity and fecundity.

## 5. Conclusions

Our experience during the implementation of OSIRIS showed us that the object-oriented programming paradigm is well suited for the creation of simulation models of ecological systems. Although mathematically very simple an individual-based model of several populations of autonomous individuals with their trophic relations, living in a spatially structured environment involves an enormous amount of bookkeeping and information exchange between the different entities. By disentangling all these entities and their relations conceptually, and implementing them as classes with ISA and HASA relations to other classes, we get a formalized, well structured and well behaving description of a ecological system. At the OSIRIS level, we want to be as general and global as possible. All basic concepts of ecological theory are implemented at this level and, at least for us, it is hard to think of an application that can not be represented in this format. At the application level however, the scope and detail of a specific

application have to be chosen carefully, with the goal of the model, the available data and the possibility to falsify the results in mind.

So far, our efforts have been limited to the implementation of models in which the environment is kept very simple or even constant, as in the *Daphnia* application. However, we think that there is a great challenge in merging the virtues of pure individual-based models with that of the more traditional production-oriented models which give a detailed description of the environment. The latter type of models have shown to be powerful in describing behaviour of an ecosystem by means of differential equations up to the level of algal dynamics (Klepper, 1989). The problem in merging both approaches lies in the way time is handled. Individual-based models often use a discrete event approach, with a separate time schedule for each individual. A system of differential equations, however, is mostly solved numerically by applying an integration routine which first calculates the derivatives of all processes and then updates all state variables. Inconsistencies may arise when individuals change their state, or any other state in the system, between the moment that the derivatives are calculated and the moment that the state variables are updated. A clever scheduling

of all the activities in the system will be needed to take full advantage of the power of what should be called individual-based compartment models.

### Acknowledgements

We thank Ramesh Gulati, Riks Laanbroek and Koos Vijverberg for their comments on the manuscript.

### References

- Baveco, J.M. and Lingeman, R., 1992. An object-oriented tool for individual-oriented simulation – Host parasitoid system application. *Ecol. Modell.*, 61: 267–286.
- Begon, M., Harper, J.L. and Townsend, C.R., 1990. *Ecology: Individuals, Populations and Communities*, 2nd Ed. Blackwell Scientific Publications, Oxford, 945 pp.
- Boersma, M. and Vijverberg, J., 1994a. Resource depression in *Daphnia galeata*, *Daphnia cucullata* and their interspecific hybrid: life history consequences. *J. Plankton Res.*, 16: 1741–1758.
- Boersma, M. and Vijverberg, J., 1994b. Seasonal variations in the condition of two *Daphnia* species and their hybrid in a eutrophic lake: evidence for food limitation. *J. Plankton Res.*, 16: 1793–1809.
- Borland International, 1993. *User's Guide Borland C++*. Scott Valey, California, 228 pp.
- Cameron T.W.M. and Billingsley, L.W., 1975. *Energy Flow: its Biological Dimensions: a Summary of the IBP in Canada 1964–1974*. Royal Society Canada, Ottawa, 319 pp.
- Chambers, R.C., 1993. Phenotypic variability in fish populations and its representation in individual-based models. *Trans. Am. Fish. Soc.*, 122: 404–414.
- DeAngelis, D.L., 1992. *Dynamics of Nutrient Cycling and Food Webs*. Chapman and Hall, London, 266 pp.
- DeAngelis, D.L. and Gross, L.D. (ed.), 1991. *Individual based Models and Approaches in Ecology: Populations, Communities and Ecosystems*. Chapman and Hall, London, 525 pp.
- Diamond, J. and Case, T.J., 1986. *Community Ecology*. Harper and Row, New York, 665 pp.
- Dyad Software Corporation, 1992. *User's Guide M++ Class Library*. Bellevue, Washington, 402 pp.
- Hogeweg, P. and Richter, A.F., 1982. INSTAR, a discrete event model for simulating zooplankton population dynamics. *Hydrobiologia*, 95: 275–285.
- Huston, M., DeAngelis, D. and Post, W., 1988. New Computer Models Unify Ecological Theory: Computer simulations show that many ecological patterns can be explained by interactions among individual organisms. *BioScience*, 38: 682–691.
- Jørgensen, S.E., Nielsen, S.E. and Jørgensen, L.A., 1991. *Handbook of Ecological Parameters and Ecotoxicology*. Elsevier Science Publishers, Amsterdam, 1263 pp.
- Klepper, O., 1989. A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (S.W. Netherlands). Ph.D. thesis, Wageningen, 270 pp.
- Klepper, O. and Rouse, D.I., 1991. A procedure to reduce parameter uncertainty for complex models by comparison with real system output illustrated on a potato Growth Model. *Agric. Syst.*, 36: 375–395.
- Knuth, D.E., 1969. *The Art of Computer Programming*, Vol. 3. Addison-Wesley Publishing Company, Reading, MA.
- Levins, R., 1966. The strategy of model building in population biology. *Am. Sci.*, 54: 421–431.
- MacMahon, J.A., Schimpf, D.J., Andersen, D.C., Smith, K.G. and Bayn, Jr., R.L., 1981. An organism centered approach to some community and ecosystem concepts. *J. Theor. Biol.*, 88: 287–308.
- Maley, C.C. and Caswell, H., 1993. Implementing *i*-state configuration models for population dynamics – an object-oriented programming approach. *Ecol. Modell.*, 68: 75–89.
- May, R.M., 1975. Biological populations obeying difference equations: stable points, stable cycles and chaos. *J. Theor. Biol.*, 49: 511–524.
- Mccauley, E., Wilson, W.G. and De Roos, A.M., 1993. Dynamics of age-structured and spatially structured predator–prey interactions – individual-based models and population-level formulations. *Am. Nat.*, 142: 412–442.
- Meijer, B., 1988. *Object-oriented Software Construction*. Prentice Hall, New York, 535 pp.
- Mooij, W.M. and Van Tongeren, O.F.R., 1990. Growth of 0+ roach (*Rutilus rutilus*) in relation to temperature and size in a shallow eutrophic lake: comparison of field and laboratory observations. *Can. J. Fish. Aquat. Sci.*, 47: 960–967.
- Mooij, W.M., Lammens, E.H.R.R. and Van Densen, W.L.T., 1994. Growth rate of 0+ fish in relation to temperature, body size, and food in shallow eutrophic lake Tjeukemeer. *Can. J. Fish. Aquat. Sci.*, 51: 516–526.
- Nisbet, R.M. and Gurney, W.S.C., 1982. *Modelling Fluctuating Populations*. Wiley, Chichester.
- Pauly, D. and Christensen, V., 1995. Primary production required to sustain global fisheries. *Nature* 374: 255–257.
- Peters, R.H., 1990. Pathologies in Limnology. *Mem. Inst. Ital. Idrobiol.*, 47: 181–217.
- Pinson, L.J. and Wiener, R.S., 1988. *An Introduction to Object-oriented Programming and Smalltalk*. Addison Wesley Publishing Company, Reading, MA.
- Polishchuk, L.V., 1992. Population Analysis and Production Approach: Two Trends in Aquatic Ecology. *Russ. J. Aquat. Ecol.*, 1: 3–8.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Ed. Cambridge University Press, Cambridge, 994 pp.
- Sarle, W.S., 1994. Neural Networks and Statistical Models. *Proc. of the 19th Ann. SAS Users Group Int. Conf.*, April, 1994.
- Scheffer, M., 1990. Simple models as useful tools for ecologists. Ph.D. thesis, Utrecht, 199 pp.
- Sigmund, K., 1993. *Games of Life: Explorations in Ecology, Evolution and Behaviour*. Oxford University Press, Oxford, 244 pp.

- Sokal, R.R. and Rohlf, F.J., 1982. Biometry, 2nd Ed. W.H. Freeman and Company, San Francisco, 851 pp.
- Starfield, A.M. and Bleloch, A.L., 1986. Building Models for Conservation and Wildlife Management. MacMillan Publishing Company, New York, 253 pp.
- Stearns S.C., 1992. The Evolution of Life Histories. Oxford University Press, Oxford, 249 pp.
- Stroustrup B., 1991. The C++ Programming Language. Addison-Wesley Publishing Company, Reading, MA, 686 pp.
- Tyler, J.A. and Rose, K.A., 1994. Individual variability and spatial heterogeneity in fish population models. Rev. Fish Biol. Fish. 4: 91–123.
- Van Tongeren, O.F.R., 1995. Data analysis or simulation model: a critical evaluation of some methods. Ecol. Modell., 78: 51–60.
- Van Winkle, W., Rose, K.A. and Chambers, R.C., 1993. Individual-based approach to fish population dynamics: An overview. Trans. Am. Fish. Soc., 122: 397–403.
- Vijverberg, J. and Richter, A.F., 1982. Population dynamics and production of *Daphnia hyalina* Leydig and *Daphnia cucullata* Sars in Tjeukemeer. Hydrobiologia, 95: 235–259.
- Wilson, W.G., De Roos, A.M. and Mecauley, E., 1993. Spatial instabilities within the diffusive Lotka–Volterra system – individual-based simulation results. Theor. Popul. Biol., 43: 91–127.
- Yodzis, P., 1989. Introduction to Theoretical Ecology. Harper and Row, New York.
- Yourdon, E., 1975. Techniques of Program Structure and Design. Prentice Hall, New York.