*Netherlands Journal of Sea Research*
*33 (3/4): 261-270 (1995)*

# SESAME, A SOFTWARE ENVIRONMENT FOR SIMULATION AND ANALYSIS OF MARINE ECOSYSTEMS

P. RUARDIJ[1], J.W. BARETTA[2] and J.G. BARETTA-BEKKER[2]

[1]*Netherlands Institute for Sea Research, P.O. Box 59, NL-1790 AB Den Burg, Texel, The Netherlands*
[2]*Ecological Modelling Centre, Joint department of DHI/VKI, Agern Allé 5, DK-2970 Hørsholm, Denmark*

## ABSTRACT

**The software environment SESAME is a model development and analysis tool designed to facilitate the construction of ecological models using Fortran-77 on UNIX machines. It consists of sets of routines which are called from a menu-driven program to perform tasks such as automatic compiling, linking and loading of a model, simulation runs, graphical and numerical output, analysis of results and comparison with available field data. It provides a choice of integration methods for simulating continuous processes and can also handle discrete processes such as encountered in systems with age- or size-structured populations. One essential feature of the package is a routine which links submodules unambiguously by automatically generating the information which needs to be common to all sections of the model. This feature, together with the concept of the distributed derivative, makes SESAME especially useful to interdisciplinary teams working on the simultaneous development and testing of complex, spatially-resolved ecosystem models comprising many modules. Some examples of models constructed with SESAME are mentioned.**

## 1. INTRODUCTION

For any aquatic ecosystem research project to be successful, it must combine and integrate the efforts of specialists in very different fields, ranging from hydrodynamics, through optical physics via nutrient chemistry to the various biological disciplines and theoretical ecology. If this integration is not accomplished, the results of such a project may still be scientifically useful for some of the disciplines involved but not for the systems ecologists who are primarily interested in the dynamic behaviour of the system as an entity and not so much in the behaviour of its constituents separately.

The active use of simulation modelling techniques, which is one of the few means at our disposal to accomplish such a testable integration, as well as enabling us to verify hypotheses about the operating principles in ecosystems and their subsystems, has been rather limited in ecosystem research. This is due to the lack of appropriate software environments that could alleviate some of the problems encountered in the construction of complex ecosystem models.

The need for simulation packages is clear, since scientists from the various disciplines who turn to ecological simulation modelling to quantify the interactions between the processes they study and other processes in the system, do not necessarily have the computer science skills and interest to handle all the aspects of translating their sets of rate equations into a functioning numerical system description and produce output that can be analysed graphically and numerically.

Many attempts have been made to cater for such a need but none of the packages created over the last 30 years are ideal for the simulation of large, spatially-resolved ecosystems. Essentially the problem is that of solving a set of parallel ordinary differential equations (ODE) as an open-ended recursive system, but the requirement for spatial resolution in three dimensions introduces the complication of partial differential equations. Conceptually the solution is to reduce the ODE's to first-order difference equations in time and to apply these within a number of discrete geographical compartments in space. Computer packages which attempt to achieve this date back to DYNAMO (Pugh, 1963), which was designed to solve problems in the field of business studies known as industrial dynamics (Forrester, 1961). In the field of ecology De Witt & Goudriaan (1974) used a Fortran-based language named Continuous System Modelling Program (IBM, 1967) to simulate a number of biological systems. The chief shortcoming of CSMP was

that it was not designed to solve partial differential equations (PDE). In spite of this it was possible to use it for simple PDE's but only for one dimension in space. This was achieved by Radford & Greenwood (1970) in modelling diffusion of gasses in soil and their uptake by microorganisms. Similarly in the GEMBASE studies (Radford & Joint, 1980) a full ecosystem model was applied to seven geographical compartments in the Severn Estuary and Bristol Channel using CSMP. A special diffusion/advection subroutine was used to calculate the transport of all dissolved and particulate matter between the compartments. Parallel to this development in Canada in the early seventies a program named SIMCON was written (Hilborn, 1973). Together with the ideas of Holling (1978) this program was further improved by the Biologische Anstalt Helgoland (Karthaus & Greve, 1983). It was written in Fortran and named BAHSIM (Fig. 1). Further features were added by the Netherlands Institute for Sea Research (NIOZ), where it was used by the BOEDE group to simulate the ecological interactions within the Ems estuary (Baretta & Ruardij, 1988) and re-named BAHBOE. Again the difference equations which described the ecology were applied to a number of geographical compartments along the axes of the Ems estuary. After considerable development at NIOZ the package was renamed SESAME (Software Environment for the Simulation and Analysis of Marine Ecosystems) at the start of the European Regional Seas Ecosystem Model project (1990). Two other simulation packages are using ideas from BAHSIM and BAHBOE, namely BSIM (Silvert, pers. comm.) and SENECA (Scholten *et al.*, 1990) (Fig. 1). Both are designed for use on PC's and stress different aspects of the modelling process, f.e. BSIM has been developed especially for portability. Besides the well-known simulation package STELLA, designed for the MacIntosh, there are other packages designed for PC's that should be mentioned, namely SIMSAB (Voinov & Akhremenkov, 1990), TIME-ZERO (Kirchner, 1989) and for UNIX environments, ELISE (Ménesguen, 1991). SIMSAB, which contains a library of functions, is intended for the construction of spatially resolved models and includes a special hydrodynamical block to calculate wind-driven currents to account for the transport between the compartments. TIME-ZERO was not developed especially for aquatic systems but it is a set of software tools to help the modeller to construct smaller models in either BASIC or Fortran by providing an interactive code generator. ELISE makes interfacing numerical output of fine-grid hydrodynamical models with box models as commonly used by ecologists very convenient.

SESAME caters to model builders of any discipline who are able to define dynamical processes as Fortran algorithms but are more than willing to leave all numerical problems of having numerous submodels interact dynamically, integrating time-varying processes with vastly different time-scales in a robust manner, and other technical problems to someone else.

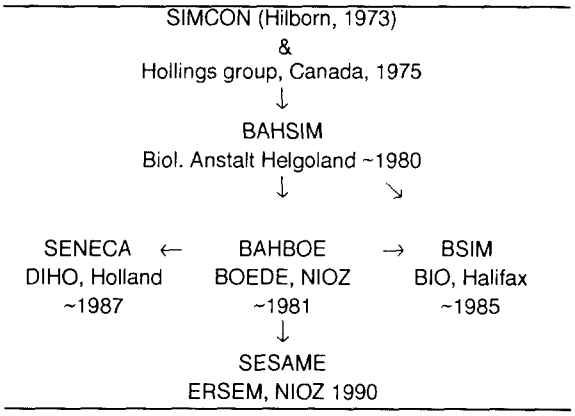


Fig. 1. The pedigree of SESAME.

The development of the package and the features it contains has always been strongly user-driven. The users, who are constructing, testing and analysing their models using the package provide continuous feedback, since the evolving models continuously introduce the need for additional features and exercise different aspects of the package.

## 2. SESAME

### 2.1. MAIN FEATURES OF SESAME

Although SESAME can be used for any simulation model, it has been developed especially for constructing complex spatially-resolved ecosystem models.

To allow the simultaneous development and testing of an arbitrary number of process descriptions, often by different experts or groups of experts, the concept of modules or submodels has been implemented. A submodel consists of one or more modules. Each module communicates with the other model components through a set of global variables, available in each of the submodels and automatically included in each module. These global variables are organized by using the COMMON-block declarations in Fortran. The advantage of this setup is that modellers can work on their own modules/submodels, while SESAME takes care of the interactions, the flows, between the modules. Another advantage is that the transport model present in any spatially-resolved model can be set up separately from the biological and chemical submodels. The division into submodels and modules is arbitrary depending on the expertise of the scientists involved and on the conceptual system to be modelled.

Acknowledging the fact that the various scientific

disciplines necessarily involved in the construction of complex models have a tendency to feel most comfortable if they are allowed to use their own dialect of formulating rate equations, SESAME allows them much freedom in this respect. However, the readability of the code requires adherence to naming conventions (Radford & Blackford, 1993) for parameter and state variable names and this demands some discipline on the part of the modellers.

In the model we distinguish three types of global variables: state variables, rate variables and other variables. The state variables have an initial value and are recalculated at every time step with the rate variables (*i.e.* integration). The rate variables are in their turn determined by the model by calculating the rate of change for each state variable. The other global variables are computed variables without initial values, whose value is calculated at each time step in one of the submodels, but may be referenced by any submodel.

Each module updates the change in the state variable(s) it defines at each time step, being aware of the state of the system through the COMMON-block which contains the values of all the global variables, and thus fully defines the state of the system. This COMMON-block is the key feature of SESAME. It links the transport submodel, which describes the transport interactions between the same state variables in different spatial boxes due to advection and dispersion, to the biological/chemical modules/submodels which describe the interactions between the different state variables in the same spatial box. It also links all the modules to the integration routine in the simulation package. Thus the COMMON block is effectively the interface between all the different parts of the complex ecosystem model.

SESAME provides the possibility to specify any variable as output variable, allowing the modeller to see the dynamic behaviour of particular variables in particular spatial compartments. It contains routines to display model results graphically, display model results and experimental data —complete with error bars— in the same plot, calculate statistical moments from the model results and allows the generation of colour-coded vertical profiles of benthic nutrient concentrations. It can display the evolution over time of a variable in all spatial compartments simultaneously as an animated map with colour-coded concentrations.

### 2.2. HARD- AND SOFTWARE REQUIREMENTS

The package was written in Fortran-77 and originally developed to run on Norsk-Data minicomputers (ND 100 and ND 500). Four years ago the package was transferred to a SunOS-UNIX machine. The package now uses the SunOS Fortran-77 compiler. The code calls subroutines, which are standard in C and as extensions on the Fortran standard. In the SunOS Fortran a number of these routines are supplied. These non-standard routines are applied in the first place to call other programs (call system) and to overcome problems related to different implementations of Fortran IO-statements in the Norsk-Data and the SunOS Fortran compiler.

Also, non-standard pointer statements, and a routine to dynamically allocate memory at run time are used to avoid limitations on use of memory and to increase the tractability of the source code.

For the graphical output of SESAME we use the SunGKS package (Version 3.0, 4.1). Because GKS (Graphical Kernel System) is an ANSI-standard, in principle any GKS-package can be used as graphical tool. We have tested it with the public-domain XGKS package.

The run time of a model run under SESAME is largely dependent on the model code, the integration method used and whether a dynamical or a fixed time step has been used. For example a model run of one year of the complex ecosystem ERSEM (version 5.2) with dynamic time stepping and using the simple Eulerian integration method, takes 600 cpu seconds on a Sun Sparcstation 10.

### 2.3. DETAILED STRUCTURE OF SESAME

SESAME provides the user with utilities to construct, compile, link and execute the model and also with extended output and analysis facilities, both graphical and numerical.

The software environment is logically divided into different parts. The most important parts contain: 1. software to compile the model; 2. software to run the model; and 3. software to analyse to model.

In addition to these software tools, the package contains tools to create a standard model and to produce templates for a new model or submodel.

#### 2.3.1. COMPILATION, LINKING AND CHECKING THE MODEL

SESAME provides a medium for the exchange of information between the submodels internally and links the model and the software environment. At compile time it generates a common block which functions as the interface between the submodels and SESAME. All state variables and other global variables are included in this common block. It is included in all submodels and in the software that controls the running of the model. SESAME produces script files with which a model can be automatically compiled and linked.

#### 2.3.2. RUNNING THE MODEL

To run the model there are

1. routines for initializing. The initial values for the state variables and the parameter values are read

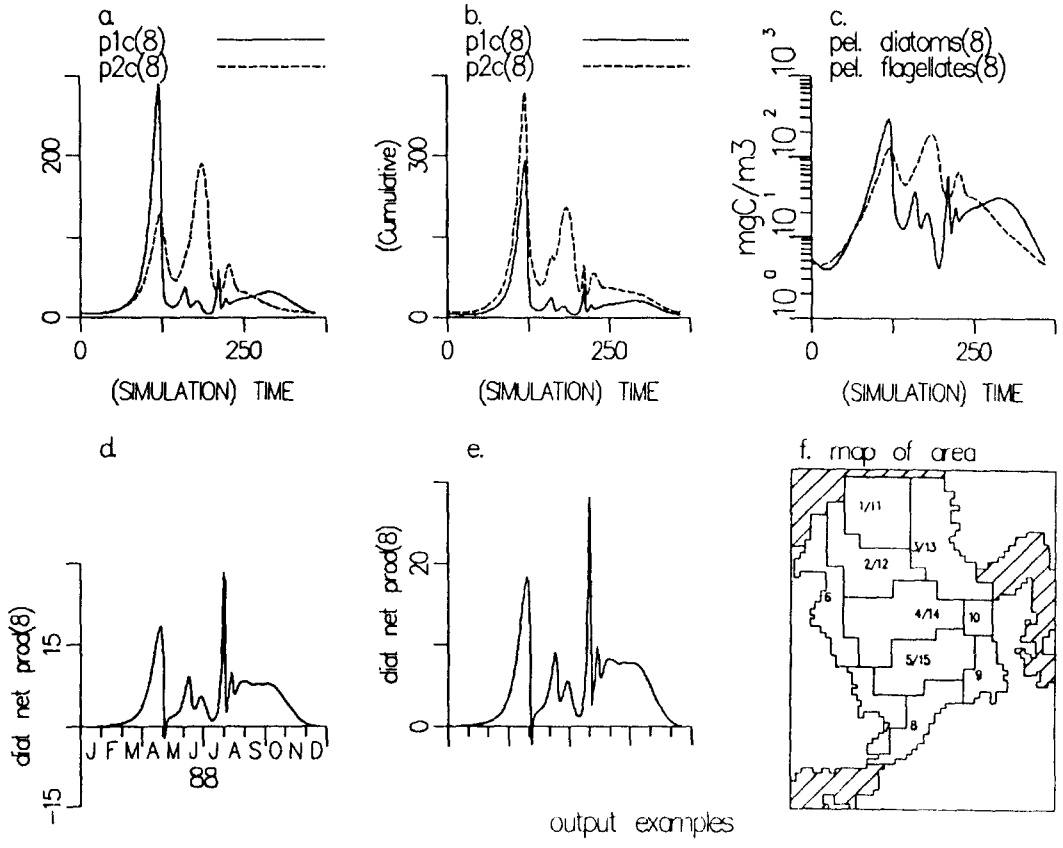


Fig. 2. Examples of output possibilities generated with SESAME. a. standard output of two state variables, with their model names. b. cumulative output of two state variables. c. output with a log axis of the same state variables with their full names and their units. d. standard output of a state variable that can have negative values. e. the same state variables with another range of the Y-axis. f. map of the modelled area. See further in the text.

with an INCLUDE statement into the working model from the data files of the standard model. These values, called the default values, can be changed in the local data files. Any changes from the original values are read in from the local data files, in order to maintain the integrity of the standard model files.

2. routines for modelling continuous and discrete processes. A model constructed under SESAME can contain two types of equations:

a. for continuous processes: a coupled set of ordinary differential equations can be solved. One can choose between two integration methods, either the Eulerian method —the rectilinear integration routine— or a fifth-order Runge-Kutta method, both with variable time stepping.

b. for discrete processes, for age- or size-structured populations, the simulation can be controlled, to ensure that special moments in time (spawning, recruitment, *etc.*) are passed exactly.

Another facility for modelling discrete processes is also included, *viz.* the ability to shift the contents or part of the contents of one state variable to another, as when moving a cohort/size class to the next class. This shifting bypasses the integration routine.

3. routines to assemble output in a file. In the model the parameter OUTDELT, the time step at which output will be stored, is given. At each OUTDELT the values of all the variables selected for output, which may be state variables, but also any other variable defined in the model, will be stored in a result file. There are two ways of storing output data: 1. the values at each point in time, which is a multiple of OUTDELT and 2. the average value over the OUTDELT period. The last one is necessary to calculate a correct budget of the fluxes through the whole system, because the simulation time step is dynamically variable, depending on the rates of change in the state variables.

4. routines to check the logical integrity of the model. SESAME checks the names of the state variables as they are defined in the standard model. In the sub-

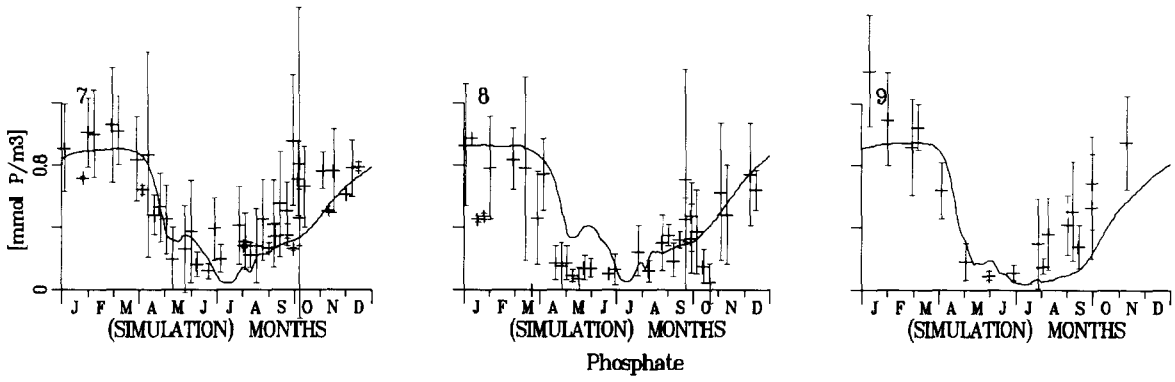


Fig. 3. Simulation output of $PO_4$ with measured values.

models no other state variables may be defined. The parameter names are checked as well. All parameters have to be defined, either in the standard model and then incorporated in the working model via an INCLUDE statement or directly in the submodel. State variables and parameters may not be defined twice and the state variable values may not be changed in the model. An error message is generated if this is the case.

5. specific routines required for particular modelling tasks, for example routines to interpolate time series.

### 2.3.3. ANALYSING THE RESULTS OF A MODEL

Analysis of results can be done numerically or graphically. The output of each model run is saved, together with an information file in which the actual parameter values and the initial values are saved, as well as other information relevant to the model run, such as year(s) of simulation, output interval, integration method, precision, etc.

SESAME has many user-definable graphical options for hard-copy output, in black-and-white or colour, on Postscript devices with a preview mode for displaying the graphs to be plotted on-screen exactly as they will appear on the hard-copy page.

Any of the graphical options found in SESAME are also available in various commercial packages but the combination of options found in SESAME has been tailor-made for use in connection with model analysis and as such is unique.

In Fig. 2a-f some of the output options are shown. The entire figure is produced by a single script file and is an example of the option to generate different types of plots in one go. The standard graphical output of two model variables has been shown in Fig. 2a. Those variables can be plotted cumulatively (Fig. 2b). This feature is useful in summing up variables, f.e. to obtain the total phytoplankton biomass, when the model results are for diatom- and flagellate-phytoplankton biomasses separately. It is possible to substitute the model names of the variables by full names, with or without units (Fig. 2c). For this purpose a file has to be made defining the full names and the units of the model variables. Model results can be rescaled, for example by logarithmic scaling to compress the range (Fig. 2c). Other possibilities of rescaling are adding up/subtracting a number, multiplication to convert to other units and exponential scaling to emphasize small differences in a seemingly featureless variable. Instead of day numbers along the x-axis as shown in Fig. 2a-c, it is possible to use months, with (Fig. 2d) or without (Fig. 2e) lettering. If the standard range of the y-axis is not acceptable the range may be redefined (Fig. 2e). If a file with coordinates is available, a map of the modelled area, with the outlines of the spatial boxes, can be drawn (Fig. 2f). Above each plot a text can be plotted (Fig. 2f), and under the entire figure a trailer can be given (Fig. 2). Other features are possibilities to draw bargraphs, polar graphs and animated bargraphs.

To analyse the results of different model runs, *i.e.* runs with different parameter values and/or initial conditions, SESAME offers the possibility to plot the results from up to three different simulations in one graph.

To help in calibrating and validating the model, experimental data may be plotted with error bars in a graph together with the results of up to three different simulation runs (Fig. 3). If ten or more data points are given, the correspondence between the data and the model results can be expressed in a figure between 0 (no correspondence) and 10 (a perfect correspondence) (Stroo, 1986).

## 3. THE ORGANIZATION OF A MODEL UNDER SESAME

In the following description one has to be aware of the distinction between the model and SESAME. The

model is the part the user defines, including the definition of the ecosystem to be modelled, while SESAME is the software environment, taking care of integration, compilation/simulation and output.

First some basic concepts of the model and the software environment are described, followed by a description of the organization of the model.

### 3.1. BASIC CONCEPTS

A spatially-resolved model constructed using SESAME, will consist of a set of descriptions of physical, chemical and biological processes in the water column and the sediment. These processes are grouped in submodels such that all physical processes are found in the physical submodel and all the biological processes and their associated chemical processes in one or more biological submodels.

Treating all the biological and chemical processes to be included explicitly in the model as independent of each other and as site-independent, only communicating with each other through the state variables allows the construction of very complex models —the most recent model developed using SESAME, ERSEM, has ~70 state variables— by an arbitrary number of modellers, all working on aspects of the same ecosystem model independently, but still receiving the necessary feedback from the other parts of the system through the state variables.

The only essential requirement to be met is that everybody uses the same units of mass for identical entities and that the aggregation of functional groups into state variables is unambiguous. The last requirement is less trivial than it sounds, since the need to reduce the huge number of (trophic) interactions in the real system always forces a strong simplification of the large number of different system constituents into functional groups, which then become the state variables.

The biological submodels are considered to be valid for the whole area modelled, so they are identical for each spatial box, or in other words they are generic. The forcing functions, such as light and temperature and the physical processes, governed by morphological properties such as depth, described in the physical submodel will be different for each spatial box, generating differences in the dynamics of the biological state variables in the spatial boxes.

In the case of a model consisting of a number of submodels, the concept of the so-called standard model is applied. The standard model is the version of the model that has been accepted by all people involved as the best current representation of the processes to be modelled. It is compiled into a library of all the submodels, and can be used as follows:

a. to combine a test version of one of the submodels with the other submodels from the standard model. To run the model only the test submodel has to be compiled instead of the whole model. This saves time especially when the model consists of many submodels.

b. to allow simultaneous developments of different submodels by different modellers. Each modeller runs his improved version of a submodel against the background of the other submodels taken from the standard model.

In this way more than one person can work on the same model: each person is working on one of the submodels and makes improvements. If the modeller(s) decide(s) that the improvements in the submodel(s) are substantial, the standard model is recompiled, the original submodels are replaced by new versions and a new standard model is established. This is an efficient and effective way of organising the effort of a group of modellers, simultaneously developing the various aspects of a complex ecosystem model.

### 3.2. ORGANIZATION OF A MODEL USING SESAME

To define a model the user has to provide SESAME with information, which must be structured in the following way. A model may include one or more submodels, with a submodel comprising at least one module. The terms submodel and module are loosely defined. The definition of the term module as we use it here is: A module is an entity that has its own datafile. Theoretically a module can be a submodel. The information for each module is located in two files, one file with the Fortran code describing the rate definitions ('module.for') and one data file with parameter values ('module.dat'). A third file must be prepared ('sesame.dat'), with control information such as initial values, integration parameters, and the selection of variables for output.

So in general there are three types of files: 'module.dat', 'module.for' and 'sesame.dat', in which 'module' may be any name. In these files different types of information are provided to SESAME. In the file 'module.for' the module proper has to be defined, with the rates as well as the boundary conditions, if the model has open boundaries. The rate definition is done by adding a calculated rate to a variable with a name consisting of the state variable name preceded by an 'S' (for Source): Sstate= Sstate + rate.

This method, the distributed derivate, is used because in complex models, where the complexity has been reduced by taking a modular approach, the rate of change is the sum of the number of terms which are calculated in different modules. The advantages of this method are discussed by Silvert (1993).

Every module uses parameters, which are constants, in the definition of process rates. The parameters with their values are defined in the file 'module.dat'. Three kinds of parameters can be distinguished: 1. single value parameters; 2. spatial parameters, which are box dependent and are defined in an array; 3. time series, which are a special type of

parameters consisting of an array of points in time. From these time series interpolated values are derived by using interpolation routines provided by the package.

The file 'sesame.dat' contains different kinds of information on the structure of the model and to control the model run. The information on the structure of the model comprises:

- the definitions and initial values of the state variables, *i.e.* the variables for which rate equations will be defined;
- the definitions of global variables for in- and output purposes and to transfer information from one module to another;
- the definition of global parameters to be used in equations in any of the modules, especially if they are to be used in more than one module.
- the definitions of sets of variables, which describe properties of the state variables such as boundary conditions or specified rates (*e.g.* transport rates). The names of these variables consist of a prefix plus the names of the state variables.

All these variables are automatically typed as reals or as real arrays. Array variables are used in spatially-resolved models, with each array element representing a spatial box.

The information on the control of the simulation run comprises

- the integration parameters, which directly control the simulation run, such as the period of simulation, the integration method, the precision, the year(s) to be simulated, the output time step;
- control information to define any discrete events during the simulation run.
- a selection of variables to store for output purposes. From all state variables and other (global) variables, defined in 'sesame.dat' or in one of the modules, output may be extracted.

Before the model can be run there must be, apart from the file 'sesame.dat' and the 'module' files mentioned before, the so-called main model, with the subroutine 'sesame-model' and with calls to the other modules.

At the start of the run SESAME converts the file 'sesame.dat' into a common block for use in all the modules and in the different routines of SESAME itself. All the 'state' and 'Sstate' variables are defined in the common block, which the integration routine of SESAME uses to calculate the values of the state variables at each time step.

## 4. EXAMPLES OF MODELS CONSTRUCTED WITH SESAME

The fundamental problem we have tried to tackle with SESAME is the problem of how to couple the physical forcing of aquatic systems by advection and turbulent diffusion with the biological and chemical functioning of these systems. We realized early in the development of SESAME that the hydrodynamics of a system largely determine the biological structure of the system and hence where in the system the major carbon and nutrient fluxes take place. The three large ecosystem models made with SESAME therefore contain a physical submodel which provides the physical forcing of the biological system.

In order to keep the computational overhead reasonable this physical forcing is provided indirectly, by aggregating the output of a general circulation model into daily advection and diffusive exchange terms across spatial box boundaries. This approach has the advantage of freeing the modellers who work on the biological aspects from having to deal with the physical aspects. However, it presupposes the availability of hydrodynamic expertise and a hydrodynamical model to generate the physical description of the system under study (*cf.* Baretta *et al.*, 1994).

Of the models constructed with SESAME or with its precursor BAHBOE, the first one was a model of the Ems estuary, a subestuary of the Dutch Wadden Sea. This is an essentially one-dimensional system, which is macrotidal, always well-mixed, and with one major axis. This allowed a simple physical submodel which was calibrated on salinity at different river discharge regimes. Constructing this model, the authors explored the importance of microbiological processes and detritus. A clear shortcoming of the model was the lack of a linkage between carbon and nutrient cycling. The Ems model has been fully documented in Baretta & Ruardij (1988).

The next model was EMOWAD, in which the biological part of the Ems model was applied to the western Wadden Sea, by forcing it with a two-dimensional physical submodel in which the exchange coefficients were generated using WAQUA, a hydrodynamical modelling system developed by the Department of Public Works (Ridderinkhof, 1988). The apparent diffusion coefficients were carefully derived from particle tracer experiments using WAQUA.

As the western Wadden Sea is mesotidal, the model structure was required to encompass an additional type of benthic subsystem, the subtidal. This is the area that does not emerge at low tide, but has much lower current velocities than the (deeper) channels and has a rich benthic fauna whereas the channels do not.

SESAME is at present being used to develop the European Regional Seas Ecosystem Model, a large modelling project of nine European marine research institutes, which is partly funded by the EC in the MAST (marine science and technology) program, partly by national funding agencies.

ERSEM is a generic model of temperate shelf seas, applied to the North Sea and simulates the seasonal cycling of particulate and dissolved organic carbon and the macronutrients N, P and Si through the food web in the water column and in the sediment. Daily advective and diffusive transport coefficients for

the ten spatial boxes, of which the five deepest are subdivided into surface and deep boxes, are derived from a general circulation model developed by Backhaus (1985; Pohlmann, 1991) and are used in the physical submodel to force the transport of the water-column state variables. Surface irradiance is climatological and box-latitude dependent. Temperature is prescribed as a sinusoid fitted to North Sea survey data with different functions for the surface and deep boxes.

Since the physical submodel defines the spatial boxes with the biological processes receiving location-dependent information from the physical submodel, the ERSEM model can be run at any spatial resolution by replacing the physical submodel.

## 5. DISCUSSION

### 5.1. TARGET GROUP OF USERS

The target group of users for SESAME comprises any combination of marine scientists who want to convert their conceptual models into a dynamic and spatially resolved form. In order to use the package for constructing their models they need a fair amount of UNIX experience, as well as a lot of general computer literacy, not to mention the ability to write Fortran code. Not surprisingly, we found that people with a strong numerical and physical background were much more likely to quickly start using this system sensibly than people with an ecological background, who have a tendency to shy away from any numerical modelling approach because 'the real systems are too complex to be modelled'. Of course they are right: to model real systems in all their splendour and complexity probably will elude us forever. However, the operative word here is 'all', for the objective of any ecosystem model is to capture the essential aspects of ecosystem function and structure, in order to ascertain whether we are beginning to understand which those essential features are. Clearly, any simulation model is a caricature of reality. As it is difficult to see a viable alternative to simulation modelling of ecosystem-level processes in interpreting and understanding those processes as they manifest themselves in their temporal and spatial variability in observations, we can only accept the limitations imposed by this method and continue to improve the realism of our models and the user-friendliness of the modelling tools used in constructing them.

Ideally, the people using SESAME to construct models should be the same people who work experimentally and/or observationally on elucidating ecological mechanisms and processes. This probably only rarely will be the case, since modelling and experimental approaches seem to be either/or activities with the additional problem that experimentalists have a tendency to believe the object of their studies to be the linch-pin of ecosystem function which necessarily must be modelled in very fine detail whereas most other components can be trivialized into a zero- or first-order function.

In collaborative modelling projects using SESAME one of the most difficult aspects was to get the domain specialists to accept that all components should be modelled at the same level of detail, which meant simplifying the functioning of some well-studied components, but also formulating and modelling hypothetical processes for lesser-known components.

### 5.2. NUMERICAL ACCURACY OF THE INTEGRATION ROUTINE IN SESAME

In order to minimize the duration of model runs the package uses a Eulerian integration method as default and runs in single precision. To satisfy the need for a very accurate integration method, a fifth-order Runge-Kutta integration routine has also been included in SESAME. The accuracy of both methods has been tested on some small models which also had an analytical solution. Both methods gave results in close agreement with the analytical result, when small time steps were used. At larger time steps there was considerable (up to 40%) discrepancy between the analytical/Runge-Kutta method and the Eulerian method. The results from complex models run with both methods only differed marginally and as the Runge-Kutta method is very slow running the ERSEM model, due to the daily varying exchanges between boxes, it is rarely used. It is quite clear nevertheless that the Eulerian method, especially for very simple models with only a few variables can produce results that can diverge significantly from the Runge-Kutta integration results. For simple models the Runge-Kutta method is the method of choice.

### 5.3. LIMITATIONS OF SESAME

A serious limitation of the package is that it has a steep learning curve and only does a limited amount of handholding. It does provide all the facilities an ecosystem modeller needs to construct, run and analyse models, but it has no facilities for gently introducing the unwary to the world of simulation modelling. It requires the user to choose the most suitable integration method for his purposes and to provide appropriate parameter settings.

Especially when building a complex model one misses the facility to have the simulation package draw a flow diagram of all the fluxes and their connections in order to visualize the dynamic structure of the model.

As the package is now, it is suitable for box models, with all the hydrodynamic information being provided in the form of (daily) exchange coefficients between horizontal and vertical boxes. It is not possible to modify the spatial setup of a model quickly, since the

necessary hydrodynamical information cannot be generated from within the package. Ideally, the package should be combined with a hydrodynamical engine, which could provide the forcing associated with water movement, at the specified spatial resolution, to the simulation model. This combination would greatly increase the flexibility of the package, but it would require still more expertise to apply it.

A fundamental requirement of any ecosystem model is that it is mass-conserving, but in the absence of a formal mass-balance check in the package which could flag violations, it is still necessary to laboriously add up all the fluxes from and to each element, in order to check a model for conservation of mass. A special routine, which stores the time-averaged fluxes, instead of the instantaneous values, is provided for this purpose. The complexity of this exercise tends to reduce the frequency with which mass-balance checks are made. Models of aquatic systems tend to involve frequent unit conversions, especially in the transition between the water column ($m^{-3}$) and the sediment ($m^{-2}$). Such conversions offer rich possibilities for errors. To assist the modellers in avoiding such errors, a simulation package should contain dimension-checking routines, especially if the development of a model is decentralized, as it is in ERSEM.

The simulation package does not impose any particular programming style upon the modellers, beyond the requirements of Fortran-77, which may be seen as a limitation from the viewpoint of code readability, since programming styles can and do diverge widely. To enhance the readability of the model code a naming convention has been adopted (Blackford & Radford, 1995), which has been adhered to very well.

## 6. CONCLUSIONS

The availability of SESAME has permitted the construction and testing of a very complex ecosystem model of the North Sea in a relatively short time by a group of modellers at different institutes.

The modellers' need to maintain a good understanding of what is going on everywhere in their models places very high demands on the analysis capability of the simulation package, which it has not been able to satisfy fully in its present form. The solution has been to create an interface in the package so the simulation results can be read in and analysed using specialized software, such as scientific visualisation packages, spreadsheets or statistical packages.

The flexibility of SESAME is best illustrated by the fact that the European Regional Seas Ecosystem Model (ERSEM), which covers the whole range of biological components from bacteria to pelagic and demersal fish, could be developed and run in it.

Acknowledgements.—This manuscript benefited from thoughtful suggestions and comments by P.J. Radford, A. Bryant and from the helpful comments of W. Silvert and an anonymous referee. The development of SESAME was partly funded by the European Union under MAST contract number CT90-0021, ERSEM.

## 7. REFERENCES

Baretta, J.W. & P. Ruardij, 1988. Tidal flat estuaries. Simulation and analysis of the Ems estuary. Ecol. Studies 71. Springer-Verlag, Heidelberg: 1-353.

Baretta, J.W., P. Ruardij, H.J. Vested & J.G. Baretta-Bekker, 1994. Eutrophication modelling of the North-Sea: two different approaches.—Ecol. Modelling **75/76:** 471-483.

Backhaus, J., 1985. A three-dimensional model for the simulation of shelf sea dynamics.—Dt. hydrogr. Z. **38:** 167-262.

Blackford, J.C. & P.J. Radford, 1995. A structure and methodology for marine ecosystem modelling.—Neth. J. Sea Res. **33:** 247-260.

De Witt, C.T. & J. Goudriaan, 1974. Simulation of ecological processes. Centre for Agricultural Publishing and Documentation, Wageningen: 1-159.

Forrester, J.W., 1961. Industrial dynamics. MIT Press Cambridge: 1-464.

Hilborn, R., 1973. A control system for FORTRAN simulation programming.—Simulation **20:** 172-175.

Holling, C.S., 1978. Adaptive environmental assessment and management. John Wiley, New York: 1-377.

IBM, 1967. Application program 1967, System 360 Continuous System Modelling Program (360A-CX-16X). User manual. IBM: 1-206.

Karthaus, V. & W. Greve, 1983. Arbeitsvorschrift BAHSIM. AVS Biol. Anst. Helgoland 1-83: 1-36.

Kirchner, T.B., 1989. TIME-ZERO: The integrated modeling environment.—Ecol. Modelling **47:** 33-52.

Ménesguen, A., 1991. 'ELISE', an interactive software for modelling complex aquatic ecosystems. In: A. Arcilla, M. Pastor, O.C. Zienkiwicz & B.A. Schreffer. Computer modelling in ocean engineering 91. Balkema, Rotterdam: 87-94.

Pohlmann, T., 1991. Untersuchung hydro- und thermodynamischer Prozesse in der Nordsee mit einem dreidimensionalen numerischen Modell. Dissertation Uni-Hamburg: 1-116.

Pugh, A.L., 1963. Dynamo user manual. MIT Press, Cambridge: 1-55.

Radford, P.J. & J.C. Blackford, 1993. The modules. ERSEM publications 1993-1: 1-250.

Radford, P.J. & D.J. Greenwood, 1970. The simulation of diffusion in soils.—J. Soil Science **21:** 304-313.

Radford, P.J. & I.R. Joint, 1980. The application of an ecosystem model to the Bristol Channel and Severn Estuary. Institute of Water Pollution Control. Annual conference.—Water Pollution Control **19:** 244-254.

Ridderinkhof, H., 1988. Tidal and residual flows in the western Dutch Wadden Sea. I: numerical model results.—Neth. J. Sea Res. **22:** 1-21.

Scholten, H., B.J. de Hoop & P.M.J. Herman, 1990. SENECA 1.2. A simulation environment for ecological application. Delta Institute for Hydrobiological research: 1-150.

Silvert, W., 1993. The distributed derivative: an aid to modular modelling.—Ecol. Modelling **68:** 293-302.

Stroo, D., 1986. A method for validation. NIOZ, interne verslagen EON 1986-1: 1-35.

Voinov, A.A. & A.A. Akhremenkov, 1990. Simulation modeling system for aquatic bodies.—Ecol. Modelling **52:** 181-205.