Modeling Environments and Applications in Ecological Modeling

W. Grossmann, H. Werthner
Institute for Statistics, Operations Research
and Computer Methods
University of Vienna

Martin Hitz
Institute for Applied Computer Science
and Information Systems
University of Vienna

Abstract
Model building for non-trivial application domains like environmental sciences is a complex task which should be carried out in a way such that resulting models not only describe the reality under investigation correctly at the chosen level of abstraction, but are also understandable, maintainable and, thus, reusable in modified experimental settings. We propose a modeling approach (supported by a computer based modeling environment) in which complex models are designed in a stepwise refinement process. Our approach is illustrated on the basis of a running example taken from forest ecology, using the conceptual framework of a modeling and simulation environment developed at our department.

Keywords: modeling environments, ecological modeling

1. Introduction

Computer based modeling and simulation serves different, however strongly related needs. On the one hand we have identification of structures, on the other hand prediction of future behavior. In practical applications both tasks are used in the context of decision support which may be understood either in a descriptive way, in order to give the analyst a picture of the system, or in an optimization mode where the analyst uses the model for finding some "optimal" decision. This means that model building has to be understood as an open loop process of stepwise refinement and restructuring of the model.

In order to achieve these goals there are different approaches and techniques available. For identification of structure one may use algebraic or graphical approaches which identifies structural relationships between variables and/or models. This has to be distinguished from the distinct relationships between the variables defined by equations. These concrete relationships between the variables are of utmost importance in the case of prediction, where a number of tools are available ranging from mathematical models over statistical models up to simulation models. Also with respect to optimization there is a broad range of possibilities from classical techniques up to new methods like neural networks or genetic algorithms.

Another important aspect in modeling is the choice of granularity. In many applications we find a spatial dimension, a temporal dimension and an individual dimension. There are two possible extreme views: at one end of the spectrum we have mathematical models formulated as a system of differential equations which allow qualitative analysis of the system behavior usually with specific information in form of parameters. In order to facilitate analysis these models are formulated frequently in a normalized way which limits the use in real

This research was supported by a grant of the Austrian Academie of Science within the IGPP-Programme.

applications. Because such models allow only prediction at a global level we call such models *global average models*. At the other end of the spectrum we have *simulation models*, used mainly by practitioners in real world situations. Models of that type are often characterized by an explicit spatial dimension and the more or less explicit consideration of individuals. Such models have frequently an abstract mathematical kernel but it is not clearly visible. Although these two approaches are linked in some sense with the two above mentioned tasks structure identification and prediction this is not necessarily the case.

In this paper we present some ideas for the structure of a modeling environment which supports the process of model building. Section 2 gives a short analysis of this process and in section 3 we show how this ideas are used in the area of modeling forest ecosystems. A modeling environment is outlined in section 4.

2. Models

2.1 Model Description

Because we are interested in models which have the opportunity to react onto the real world our main interest are dynamic input–output models as defined by Zeigler (1976). Formally such models have a set of input variables, a set of parameters, a set of output variables together with associated output transformation functions, a set of state variables together with their state transition functions and a definition of the time resolution. Usually we have not a simple model but a rather complex structure which is defined in the process of model derivation which means coupling of different model components. As a result we obtain a model graph where the nodes are the model components, represented as boxes and the arcs define the dependency between the components. From a formal point of view we distinguish different types of models:

Basic model: This type of model is described by the attributes enumerated above and does not contain any submodels.

Static model: By this term we denote a special kind of basic model characterized by the fact that no state variables are used. Such models are usually employed as auxiliary nexus in the process of model coupling.

Compound model: This model type is obtained by coupling models of any kind. Such coupling is done by connecting input and output variables of the components models.

Multimodel: In order to describe discontinuities in the behavior multimodels are used (cf. Zeigler et al., 1993). A multimodel consists of structurally (in terms of input and output structure) compatible component models with distinct behavior which play a mutually exclusive role during simulation.

Array model: An array model can be thought as a special kind of compound model consisting of a varying number of components of a certain type arranged in a parallel fashion. The input data are distributed to each component by an input fork function and the output of the model is calculated by an output aggregation function from the component outputs. Moreover there exists a regeneration and a destruction function in each array model which allows the control of the number of components.

All these model categories may be organized in an object oriented class hierarchy which encompasses also different methods of computation (Grossmann et al., 1996). In applications it is important to have a unified description for all types of models which is given by the following attributes (cf. Schwab et al., 1997):

Name - describing the main purpose of the model
Extensive textual description - explaining the main characteristics
Model type indicator - indicating the model type in the sense defined above
Model principle - describing the general performance of the model
Usage level - indicating whether this model may be applicable inside an array model
Internal structure- giving the information about the submodels contained inside the model together with the coupling information
Input variables - consisting of name, textual description, measurement unit and value range of the input variables
Output variables - consisting of name, textual description, measurement unit and value range of the output variables
State variables - consisting of name, textual description, measurement unit and value range of the state variables together with the initial values
Internal variables - auxiliary variables used for internal calculations
Parameters - used in the calculation of the state transition function
Time resolution- used for calculation of the state transition function
State transition function - updates the state variables
Simulation method - a numerical procedure for solving recursive algebraic equations in case of difference equations or differential equation in case of continuous systems.

Note that not all these attributes are meaningful for all types of models, for example, in case of static models the state transition function is not meaningful. During the process of model derivation the most important attributes are the *Model principle*, the *Internal structure* and the *Usage level*. The *Model Principle* may be seen as the basic idea behind the state transition function for the state variables of primary interest, no matter what internal complexity the model has. Typical examples are population balance equations of the form

$$\text{New Population} = \text{Old Population} + \text{Birth} - \text{Death}.$$

Sometimes such *Model principles* may be captured in a precise form by a standard differential equation model like a saturated growth model, a predator prey model or a cohort population model (Leslie model). Note that such principles say nothing about the internal complexity of a model. For example a cohort model based on the Leslie principle model may consist of a number of models which represent the cohorts and may consider the population mass as well as the number of individuals in each cohort as state variables, whereas a saturated growth model consists only of a single equation for a single state variable.

The *Internal structure* of the model is of utmost importance for compound models. Corresponding to the *Model Principle* we distinguish several layers in the description of the internal structure. The *primary layer* contains all model components which formalize the *Model Principle*. The *secondary layer* describes model components which are used in the calculation of auxiliary quantities, frequently used by the main components, for example as substitution of the parameters. Independent from the layer each model component of a compound model needs information about the connection to other model components.

The *Usage level* of the model tells us whether a model may be used at the aggregate level, at the individual level or at both levels. This distinction is important for the definition of array models. Only models with an individual *Usage level* can be used as components of an array model.

As an example consider the compound model shown in figure 1. The modeling principle is defined by the generic cohort model (or Leslie model). Corresponding to this *Model principle* the primary components of the model are the components representing the cohorts denoted by *Age Class i*. The two models *Food* and *Harvesting* are at the secondary layer and are used for the calculation of the time varying parameters in the primary components. The *Food* component is a dynamic model whereas the model *Harvesting* is a static model represented in the figure as a box with double boundary. This interpretation is only one possible way to understand this model. Another one would be to include the food model into the primary model. With respect to the *Usage level* we assume that this model can be used only at the aggregate level because we have in the standard interpretation a percentage for each cohort which moves to the next age class.

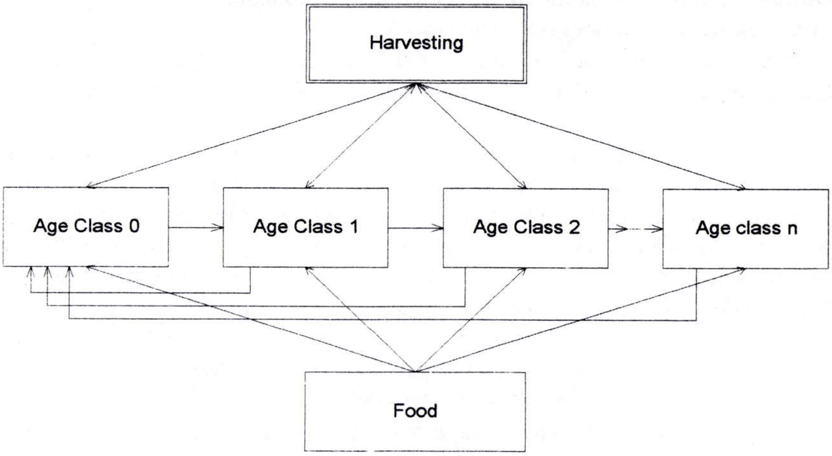

Fig. 1: Cohort model with secondary components

2.2 Model Derivation

In the process of model derivation the model builder uses either a top down or a bottom up approach. In the top down approach one starts with a generic *Model principle* and derives in several refinement steps the final model, in the bottom up approach one starts with a number of well defined model components and aggregates them to a new model. In any way the resulting new model has to be documented in an appropriate way.

Looking at the process of model derivation from a more applied point of view we can distinguish two types of derivation. The first one is more oriented towards structure, the second one more towards refinement.

2.2.1 Structural Derivation

By structural derivation we understand the introduction of a new model component of any type except an array model. In this case first of all the linking of the new model component to the already existing model has to be defined. This is done by defining the outputs of other models which are inputs for the new component and those already existing model components for which the outputs are used as inputs. We distinguish the following two types of connections:

(1) *Direct connection*: Inputs and outputs are connected with outputs and inputs of already existing model components.
(2) *Indirect connection*: Inputs and outputs are connected to already existing models in another way, for example as a substitution of parameters or as new variables in the state transition function.

The first case is well defined and rather easy to handle. More intricate is the second case because a *connection principle* has to be identified and there may be a number of side effects which have to be considered. This side effects have to be documented in the *Model principle* and in the *Internal structure* of the model. Note that the principle may change by the connection principle. This *Model principle* is of utmost importance in the automated definition of the computation. Take for example the model of figure 1. If we want to derive such a model from scratch in a bottom up manner we would start with a single model component for the entire population. This model is a very simple model defined by the *Model principle*: *proportional growth with loss* and the state transition function is given by

$$X(t + \Delta t) = X(t) + [\lambda X(t) - \delta X(t)]\,\Delta t$$

The parameter λ represents the growth rate and δ gives the mortality rate. Time increment is defined by Δt. This model may be interpreted as a growth model for a homogenous population without any regeneration. In order to make such a model more realistic we derive from this model in the next step a model with two classes (children and adults) as it is depicted in figure 2 and which is described by the following dynamics:

$$X_0(t + \Delta t) = X_0(t) + [\lambda_0 X_0(t) - \mu X_0(t) + c\, X_1(t) - \delta_0 X_0(t)]\,\Delta t$$

$$X_1(t + \Delta t) = X_1(t) + [\lambda_1 X_1(t) + \mu\, X_0(t) - \delta_1 X_1(t)]\,\Delta t$$

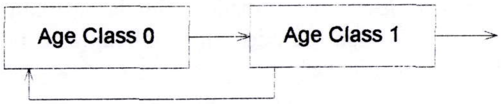

Fig. 2: Simple cohort model with 2 Age classes

The parameter c represents the regeneration rate of the adults, the parameter μ represents the dynamics of transfer from children to adults and the parameters λ_i and δ_i represent cohort specific growth rates and mortality rates. This coupling is of course of indirect type because we have used a special interpretation of the input of the new component. Due to this coupling we have also a change in the modeling principle from *proportional growth with loss* to the *age dynamics of cohorts*. Proceeding further by introducing more age classes we keep the modeling principle but obtain only a change in the *Internal structure* and arrive at the *primary components* of figure 1. The introduction of the components for harvesting and food is also an indirect coupling of the components because both components affect only the transition parameters μ, δ and λ.

2.2.2 Disaggregation and Aggregation

Under this heading we summarize all considerations in connection with spatial and individual structure. A model obtained in the derivation process may fall with respect to this spatial and individual dimension into one of the four categories defined by the two dichotomies:

- models at the individual level versus models at the aggregate level
- models with explicit spatial dimension versus models without spatial dimension

In order to switch to another category we use derivation steps characterized by the terms aggregation and disaggregation.

Models at an individual level may be easily aggregated to a population level by using an array model component. Moreover the input fork function allows to some extent the consideration of competition between different individuals, in particular all kind of competition which is competition for input variables in dependence of the state of the output variables. Using the regeneration and destruction mechanism of the array model we can also define population dynamics at an individual level, similar to the case of discrete event simulation.

A little bit more care is needed in case of disaggregation of population models to an individual level. In that case the value of the *Usage level* of the model is of utmost importance which states whether the model allows disaggregation or not. The crucial point is how the model at the population level handles competition, regeneration and destruction. As long as such considerations are not included into the model it is often possible to make disaggregation. In case of implicit modeling of such phenomena disaggregation will need a major restructuring of the whole model and consequently we would not speak from model derivation.

With respect to the spatial dimension both processes are usually easy to handle. The typical example is consideration of grid cells. A model for a well defined area can be transferred easily to larger areas by using an array model with appropriate input variables. Note that this array model needs no regeneration or destruction function. Similar we can do disaggregation by specification of new inputs for a number of grid cells. Difficulties may arise also in this case mainly from interactions between spatial cells.

2.3 Model computation

In order to compute the model one has to define an executable program from the graphical structure set up during model derivation. Various attempts in this area have already been made and at the moment it seems that the most efficient process is that one which defines a compiled version of the program. The process of generating this program depends on the model which is defined. From a formal point of view every model which has to be executed can be understood as a compound model with a final output and an initial input. The derivation of the program for a compound model depends on the *Internal structure* of the model. Basically we have to distinguish three cases.

The first one is a compound model consisting only of static components. In that case computation reduces to one formula which can be derived by tracing the graph backwards from the final output to the initial input and finding in this graph the longest path, i.e. the path which contains the maximum number of model components. Note that in this case the model can only be calculated if this graph is acyclic.

The second case is the standard application of a compound model with dynamic and static components. The dynamic components may be formulated as differential equations or difference equations and can be solved by any numerical method for solving differential equations. In most applications Euler's method is used but also more complicated procedures are possible. All these methods need the definition of an appropriate time increment. This may be difficult in case of models with different time resolutions but let us assume at the moment that the model is homogenous with respect to the increments. Again the sequence of computation is defined by the *Internal structure* of the model. First of all the secondary components are updated and afterwards the primary components. Within each block we have *a priority* rule between static and dynamic components. First the static components are updated according to the principles stated above. Next the dynamic components are updated by the following formula for the state transition functions which resolves automatically feedback loops :

New value = f(Old values of state variables, Input variables, Parameters)

If we look at the example of the generalized cohort model from figure 1 we would obtain the following basic structure for the program which is given in some detail in Hitz et al. (1997):

```
Update Food
Update Harvesting
Update Age Classes
```

The third case is the update for array models. In that case the basic steps are given by

```
Input aggregation function
Deastruction function
Loop for state transition function of model components
Regeneration Function
```

This sequence reflects to some extent the biological logic that regeneration at the individual level is subordinated to survival represented by the state transition function. In case of nested array models the procedure may be generalized in that way that we put inside the inner loop for the state transition another array model component. Details may be found in Grossmann et al., 1997.

3. Application of model construction

Because we have already considered the case of model construction in the bottom up manner to some extent in the previous section we will consider in this section only the case of top down model building. As an application area we will use forest simulation models. The basic principle for all vegetation models is the idea that we have to consider birth death and growth inside the forest. Moreover most practitioners agree that a model should consider individual trees. In particular the following features are considered as the most important ones (cf. Bossel, 1991)): Productivity of trees, growth dynamics, competition, birth and death, individual characteristics like shape, local site effects. In order to construct such a model we can start from a generic model as it is shown in figure 3

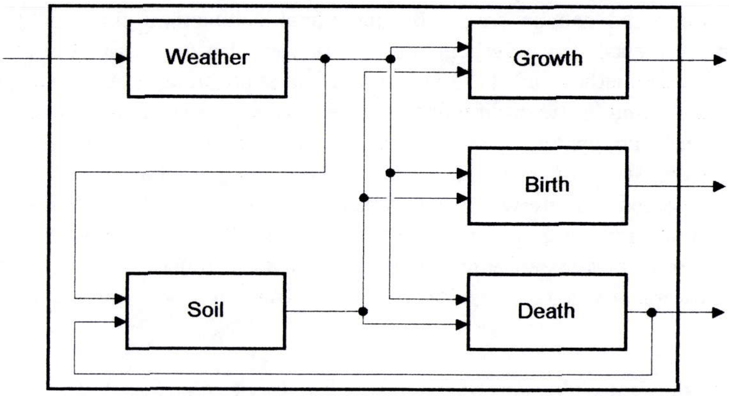

Fig. 3: Generic growth model

In order to obtain a concrete model one starts in many cases with a growth model for the individual tree. Let us consider here only the simplest case defined by a saturated growth model of the following form:

$$\Delta V = \gamma \cdot LA \cdot (1 - ((D \cdot H) / (D_{max} \cdot H_{max})))$$

Here D denotes the diameter, H the height of the tree and D_{max} and H_{max} denote the maximum height of the tree. The variable LA is thought as the leave area which is assumed to be proportional to the volume of the tree, which is a function of diameter and height. Hence we can assume that volume or equivalently biomass is the state variable and all the other variables are auxiliary variables.

In the next step we add to this model input data which are given by temperature, precipitation light and soil conditions. In order to incorporate these inputs into such a simple model a standard technique is the definition of factors for each input which modify the growth equation. Because we assume that soil conditions are rather stable over time we represent soil conditions as a parameter. The other quantities are time dependent and are calculated by static model components. The most detailed model is in many cases that one for light because we assume that competition for light is the most important interaction component between trees. One way to define such a model is to assume that the light input is a vector of available light in different heights above ground. From these values we calculate the light factor using the leaf area in different crown layers. Putting all these models together we obtain the model for a single tree as it is shown in figure 4.

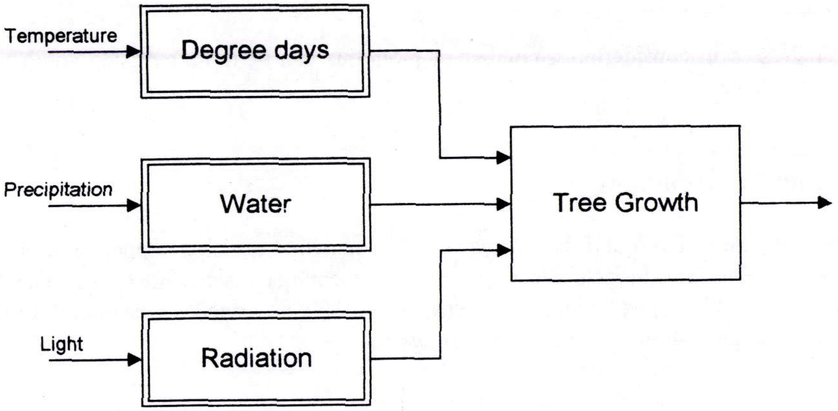

Fig. 4: Tree model for a single tree

The *Model principle* of the whole has now changed and is defined as *saturated growth dependent on external conditions*. The update of the state variables is given by the following equation:

$$\Delta V = \theta_{soil} \cdot \theta_{light} \cdot \theta_{temp} \cdot \theta_{water} \cdot \gamma \cdot LA \cdot (1-((D \cdot H) / (D_{max} \cdot H_{max})))$$

Next we use an aggregation step which allows consideration of a number of trees on a small area called plot. This is done by putting a number of tree models into an array model. The crucial step is of course the definition of competition between trees. Again we consider only competition for light between individuals which is given due to the fact that smaller trees are shaded by the crown of larger trees. In order to represent this fact in the model we use a static model which calculates the available light in different heights above crown from the overall leaf area on the plot. This overall leaf area on the plot defines an additional output variable for the array model. Graphically the model can be represented as shown in figure 5. The box *Trees* represents the array model and the *Model principle* has now changed to T*ree community growth with light competition for individuals*. This model is the classical gap model formulated in Shugart (1984), a recent implementation may be found in Urban (1990).

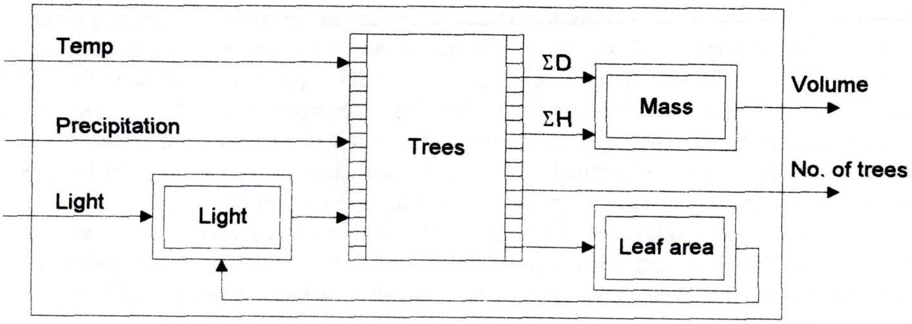

Fig. 5: Community growth model with light competition

If one is interested in considering a number of plots this can be done by using another array model for the plots.

4. Modeling Environment

A number of authors have found that in case of environmental decision support a coordinated interaction between data, models and representation techniques is necessary (cf. Van Voris, 1993). A general architecture for modeling environments is described for example in Guariso et al. (1996). The general outline is shown in figure 6.

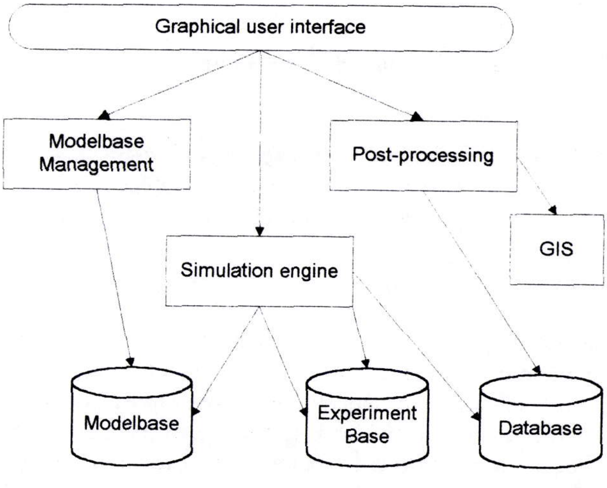

Fig. 6: Modeling environment

There are three databases which are used in the environment: a model base, an experiment base and a data base.

The model base contains all models developed up to now which are represented as objects with attributes as described in subsection 2.1. Besides the structuring given by the object representation, the model base may have an additional structure given by the knowledge in the ground domain. In the area of forest modeling such a structure is defined by the generic model represented in figure 3 and we may distinguish between a model base for growth, one for competition, one for regeneration and mortality and one for transformation of input data. The model base for growth contains mechanistic models for single trees which vary with respect to the underlying physiological assumptions, but also models defined at the population level are included. The model base for competition offers models for nutrient and light competition. Light competition models are characterized by geometric concepts for tree crowns which may be formulated in different ways. Also models for nutrients use geometric concepts, but for the root area. The model base for regeneration and mortality offers various models for birth, usually formulated at the community level, whereas mortality may be formulated on the community as well as the individual level. The model base for input transformations offers the opportunity to transform input data from the data base into a form appropriate for use inside the other models.

The data base contains time series data as well as static data. All these data may be either monitoring data as well as data obtained by simulation results. Each entry in the data base may be characterized by the following aspects (cf. Plank, 1994):

Metric Scale Aspect: describes the metric scale of the measured value, which can be percent, parts per million, or standard metric units.

Temporal Aspect: specifies the date and time of the measuremment.

Spatial Aspect: describes the location, where the corresponding measuring station is installed. Many spatial (topological) relationships may arise form this spatial aspect of any measured value.

Thematic Aspect: holds information of the particularities of the value. It may include information about the reliability of the measured value, the method of measurement or the provider of the data set.

Structural Aspect: concerns the time series structure like regular or irregular measurements.

The experiment base is orthogonal to the model base and administers experiments defined by model instances, together with initial conditions, parameters and input data. The main goal of this data base is to facilitate comparison of results obtained by models is coupled with different data and vice versa.

The most important components for processing are the model base management, the simulation engine and the post-processing tool. The model base management is responsible for handling the model contained in the model base, i.e. retrieval, update and storage. It supports reusability of models by maintaining additional structural and semantic relationships between models, for example classification of models according to use in the ground domain as stated above, which may be exploited for retrieval purposes by an interactive model description browser which offers selected attributes as well as graphical model representation.

The simulation engine is able to handle heterogeneous models consisting of submodels which may be based on difference or differential equations. The basic method for the simulation engine is outlined in section 2.3.

The post-processing tool allows comparison of different experiments as well as the presentation of the results using GIS facilities and statistical tools.

The graphical user interface supports modeling without programming. It includes a form based model editor, a graph based compound model designer and a model search engine.

5. Conclusions

We discussed several modeling principles within the domain of ecological modeling. However, we think that these principles are generally applicable. Moreover, we presented a modeling environment, its architecture is the outcome of several prototypes developed and implemented at our institute.

References

Bossel, H. (1991). Modeling Forest Dynamics: Moving from Description to Explanation, Forest Ecology and Management 42, p.129.

Grossmann, W., Werthner, H., Hitz, M. (1996). An Object-Oriented Simulation Environment Supporting Self-Adapting Model Structures and Qualitative Models, Sixth Annual Conference on AI, Simulation and Planning in High Autonomy System, California, USA, p. 125.

Grossmann, W., Werthner, H., Hitz, M. (1997). Parallelism in Continuos System Simulation. Proceedings of the 1997 Western Multiconference, OOS ´97 (J. Wallace, T. Beaumariage, Y. Dessouky) p. 25 - 30.

Guariso, G., Hitz, M.& Werthner, H. (1996). An Integrated Simulation and Optimization Modeling Environment for Decision Support. Decision Support Systems, No. 16, pp. 103 – 117.

Hitz, M., Werthner, H. (1997). Earning Benefits of the Object-Oriented Pardigm in Dynamic System Simulation. Proc. of the HICCS 1997, Hawaii, January 1997.

Plank, T. (1994). EQUEL – A Geographical Information System for the Management of Spatio-Temporally Distributed Data. Dissertation, University Vienna.

Shugart, H.H. (1984). A Theory of Forest Dynamics. Springer, New York.

Schwab, M., Werthner, H., Guariso, G. (1997). Data Structures in Environmental Modeling. To appear in Proceedings IMACS ´97

Urban D.L. (1990). A versatile model to simulate forest pattern – Users guide to ZELIG. Environmental Science Department, University Virginia Charlottesville.

Van Voris, P. (1993). TERRA-Vision – The integration of scientific analysis into the decision-making process. Internal Journal of Geographic Information Systems 7(2) 143 - 164.

Zeigler, B. P. (1976). Theory of Modeling and Simulation. John -Wiley, New York.

Zeigler, B. P., Louri, A. (1993). A Simulation Environment for Intelligent Machine Architectures. J. on Concurrent & Distributed Processing.