Jul 1st, 12:00 AM

Modular Ecosystem Modeling

Alexey Voinov

Carl Fitz

Thomas Maxwell

Roelof Boumans

Robert Costanza

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

Voinov, Alexey; Fitz, Carl; Maxwell, Thomas; Boumans, Roelof; and Costanza, Robert, "Modular Ecosystem Modeling" (2002).
International Congress on Environmental Modelling and Software. 117.
<https://scholarsarchive.byu.edu/iemssconference/2002/all/117>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Modular Ecosystem Modeling

Alexey Voinov¹, Carl Fitz², Thomas Maxwell¹, Roelof Boumans¹, Robert Costanza¹

¹ University of Maryland Institute for Ecological Economics, P.O.Box 38, Solomons, MD 20688
(voinov@cbl.umces.edu)

² South Florida Water Management District, Everglades Department

Abstract: The modular ecosystem modeling approach was used to create a flexible landscape model structure that is easy to modify and extend for particular case studies and applications. The Library of Hydro Ecological Modules (LHEM; <http://iee.umces.edu/LHEM>) includes modules that describe hydrologic processes, nutrient cycling, vegetation growth, decomposition, etc., both locally and spatially. LHEM is implemented within the framework of the Spatial Modeling Environment (SME; <http://iee.umces.edu/SME3>) that integrates modules and places local simulation models into a spatial context. The LHEM was used to build the Patuxent Landscape Model (PLM) as well as models of several subwatersheds of the Patuxent. Local ecosystem dynamics are replicated across a grid of cells that compose the rasterized landscape. Different habitats and land use types translate into different parameter sets that drive the modules chosen. Spatial hydrologic modules define horizontal fluxes of material and information and link the cells together. Results show good agreement with data for several components of the model at several scales. The modular structure was essential to refocus the model on several problems associated with different management decisions that were to be made.

Keywords: Landscape modeling; modularity; scaling; dynamic spatial modeling.

1. INTRODUCTION

As we tackle more complex systems in our studies, complexity of the models we build also tends to increase. At some point we may end up with models which complexity becomes prohibitive for proper analysis and understanding: we substitute one complex system, - the system we study, by another complex system, - the model we built. The complexity in the model structure makes it hard to understand the linkages and logic, the volume of output generated (especially in spatial models) becomes more than we can analyze and interpret. In many cases we tend to rely on innovative software that will be there to handle all the complexity. However certain problems are better handled methodologically, rather than just mechanistically. We argue that the modular approach can be helpful to deal with many complexity issues, and is especially important in the research domain, where models are primarily built to further our understanding of systems.

2. MODULARITY

In the *modular* approach instead of creating a model general enough to represent all the variety of ecological systems under different environmental conditions, we develop a collection of modules simulating various components of ecosystems or entire ecosystems under various assumptions and resolutions. In this case the challenge is to put the modules together, using consistent and appropriate scales of process

complexity, and make them talk to each other within a framework of a full model. The concept of modularity gained strong momentum with the wide spread of the object oriented approach in software development [Silvert, 1993; Sequeira et al., 1997].

Reynolds and Acock [1997] offer an extensive discussion of modular design criteria and rules in application to plant modeling. The features of *decomposability* and *composability* are probably the most important ones. The decomposability criterion requires that a module should be an independent, stand-alone submodel that can be analyzed separately. On the other hand the composability criterion requires that modules can be put together to represent more complex systems. Decomposability is mostly attained in the conceptual level, when modules are identified among the variety of processes and variables that describe the system. There is a lot of arbitrariness in choosing the modules. The choice may be driven either by purely logical, physical, ecological considerations about how the system operates, or by quantitative analysis of the whole system, when certain variables and processes are identified as rather independent from the other ones.

The composability of modules is usually treated as a software problem. That aspect is usually resolved by use of wrappers that enable modules to publish their functions and services using a common high-level interface specification language (the federation approach) [CORBA, 1996; Villa and Costanza, 2000]. The other

alternative is the design of model specification formalisms, that draws on the object-oriented methodology and embeds modules within the context of a specific modeling environment that provides all the software tools essential for simulation development and execution (the specification approach) [Maxwell, 1999]. In both cases, as models get into the hands of software developers, the gap between the engineering and the research views on models and on their uses starts to grow. From the software engineering viewpoint the exponential growth of computer performance offers endless resources for the development of new modeling systems that can take care of any complexity. With the advent of the Internet it becomes possible to assemble models from building blocks connected over the Web and distributed over a network of computers [Fishwick et al., 1998]. New languages and development tools appear even faster than their user-communities manage to develop.

On the other hand from the research viewpoint, if a model is to be a useful simplification of reality it should enable a more profound understanding of the system of interest. It is more important as a tool for understanding the processes and systems, than for merely simulating them. In this context there is a more limited demand for the overwhelming complexity of modeling systems. The existing software may remain on the shelves if it does not really help understand the systems. This is probably especially pertinent to models in biology and ecology, where in contrast to physical science or engineering, the models are much more loose and “black-box” much of the underlying complexity due to the difficulty of parameterizing and simulating all of the mechanisms from a first-principal basis. They will require a good deal of analysis, calibration and modifications, before they may be actually used. In this case the focus is on model and module transparency and openness. For research purposes it is much more important to know all the nuts and bolts of a module to use it appropriately. The “plug-and-play” feature that is so much advocated by some software developers becomes of lower priority. In a way it may even be misleading, creating the illusion of simplicity of model construction from prefabricated components, with no real understanding of process, scale and interaction.

3. THE SME/LHEM APPROACH

Instead of a general model that should represent all the variety of ecosystems, we can formulate a general modular framework (Fig. 1), which defines the set of basic variables and connections between the modules. To provide transparency, whenever possible, we formulate modules in terms of STELLA [HPS, 1995], an icon-based modeling system. Particular implementations of modules are

flexible and assume a wide variety of components that are to be made available through libraries of modules. The modules can be developed, tested and used independently. However they can share certain variables that are the same in different modules, using a convention that is defined and supported in the library specification table. When modules are used as standalone components, these variables are specified by user-defined constants, graphics or timeseries.

To bring the modules together we use the Spatial Modeling Environment (SME) [Maxwell and Costanza, 1997] that can take individual STELLA modules and translate them into a format that supports modularity. In addition to STELLA modules, SME can also incorporate user-coded modules that are essential to describe various spatial fluxes in a watershed or a landscape. Within the SME context the shared variables from different modules get updated in other modules to create a truly dynamic interaction.

Figure 1. LHEM local and spatial modules. The local modules are formulated as STELLA models; the spatial modules are C++ code.

For more complex processes and interactions, such as the ones describing spatial dynamics, modules can be formulated in C++. They can use some of the SME classes to get access to the spatial data and can be then incorporated into the SME as Usercode, and used to update the local variables described within the STELLA modules. In case of Usercode it is hard to offer the same level of transparency as with the STELLA modules. More emphasis should be made on explicit documentation and comments to the code. We also hope that by presenting the various modules of the LHEM on the web and offering detailed description we can increase their utility for reuse and further improvement. However we do not want to suggest any mechanical incorporation of these modules into any future models. They are only samples and blueprints of modules for other modeling efforts.

There is a good variety of software currently available that can help build and run models. Between the qualitative conceptual model and the computer code, we may place a number of software tools that can assist us in converting

conceptual ideas into a running model. Usually there is a trade off between universality and user-friendliness [Voinov and Akhremenkov, 1990]. On the one extreme we see computer languages that can be used to translate any concepts and any knowledge into working computer code. On the other extreme we find realizations of particular models that are good only for the individual systems and conditions that they were designed for. In between there are a variety of more universal tools.

They include modeling languages, which are computer languages designed specifically for model development, and extendible modeling systems, which are modeling packages that allow specific code to be added by the user if the existing methods are not sufficient for their purposes. In contrast, there are also modeling systems, which are completely prepackaged and do not allow any additions to the methods provided. There is a remarkable gap between these packaged and extendible systems in terms of their user-friendliness. The less power the user has to modify the system, the fancier the graphic user interface and the easier the system is to learn. From modeling systems we go to extendible models, which are actually individual models that can be adjusted for different locations and case studies. In these the model structure is much less flexible, the user can make choices from a limited list of options and it is usually just the parameters and some spatial and temporal characteristics that can be changed.

Similarly for modeling environments such as the SME there is a certain level of user-friendliness that is usually in reverse proportion to generality. To be able to link both unit and spatial modules together, SME adopts certain conventions on how the modules should be described and what are the formats of data that can be used.

For SME, a module should be described as a sector in STELLA. The variables within a sector will be considered as owned by this module. All the external variables that are defined outside of the sector borders can be defined in other modules. Within a module, to make it operable as a stand-alone model, these external variables should be defined as constants or as timeseries (say, defined as graphs in Stella) that can change with time or as functions of some other independent variables.

Variables that are shared between modules should have the same name. The SME translator takes the STELLA equations saved as a text file, and translates them into an intermediate formalization, called the Modular Markup Language (MML). It will find the shared names and link them together. A *config* file will be produced that contains all the variables from all the modules. This *config* file can be further edited to change the values used for the variables in the driver.

As local dynamics get treated in the SME in a spatial context, it also gets the spatial variability that can be associated with the various parameters being spatially distributed, related to, say, soil or habitat types. In this case when moving from one spatial locality to another the same system of ODE's generated from STELLA gets to be solved with a different parameter set, one that is substituted by SME. Currently SME does not incorporate any extensive data base features to serve the needs of describing and archiving the numerous parameters encountered in models and modules. However there are several well-elaborated input mechanisms that allow one to read the location-dependent data from various file formats. For example, the habitat-dependent parameters are accumulated in a file that has various columns representing the different model parameters, and rows describing the various habitats. A parameter described as habitat-dependent in the *config* file is then input from this file based on the information about the particular habitat specified by the Land Use map.

4. CALIBRATION AND TESTING

We have been mostly using the LHEM for modeling of the Patuxent watershed as well as several of its subwatersheds. Another watershed that was modeled is the Gwynns Falls, a highly urbanized watershed in Baltimore. This brief description of the application of the LHEM in a particular project is primarily to illustrate how the modules were assembled and calibrated. The details of the PLM and its results have been reported elsewhere [Voinov et al., 1999; Costanza et al., 2002].

The modular approach called for the decomposition of the calibration process into what we termed a multi-tier calibration method (Fig. 2). In this case the calibration of the full model has been achieved in a step-wise process that started with the calibration of individual modules. The obvious benefit of this was a much simpler model to calibrate at each step. It is also clear that the aggregate of several modules does not necessarily behave similarly to the individual modules taken separately. Therefore recalibration was needed every time we went from simple individual modules to their combinations, both locally and spatially. However, it was always much easier to fine-tune the already performing modules, than to do a full-scale calibration of the full model in its overall complexity.

We started with calibrations of the local hydrologic model. The input from other modules, primarily plants was imitated by fixed time series. For example, a time series was generated to represent the approximate dynamics of plants over a one-year time period. The model produced clearly different dynamics with and without plants. This

stage of local calibration is important to make sure that all the major fluxes, such as evaporation, transpiration, percolation, nutrient loading, and uptake, are within some reasonable values, that there are no inexplicable trends in the model. It was also important to run the sensitivity analysis and understand the effect of individual parameters on overall dynamics.

Figure 2. The process of multi-tier modular calibration. The additional recalibration needed when switching from one spatial or structural scale to another is well compensated by the simplicity of calibrating smaller and simpler modules.

Most of the calibrations for these modules were done in combination with the module for spatial dynamics added within the SME context. The spatial model for hydrology and nutrients, or the water quality model has been calibrated for several subwatersheds in the Patuxent. We identified two spatial scales at which to run the model - a 200m and 1 km cell resolution. The 200m resolution was more appropriate to capture some of the ecological processes associated with land use change but was too detailed and required too much computer processor time to perform the numerous model runs required for calibration and scenario evaluation. The 1km resolution reduced the total number of model cells in the watershed from 58,905 to 2,352 cells.

We did most of the preliminary calibrations for a small (23km²) subwatershed of Cattail creek in the northern part of the Patuxent basin. Another small subwatershed that of Hunting Creek, was located in the southern part of Patuxent, quite different in terms of soils and elevations. It also included an estuarine part that allowed us to test the second hydrologic algorithm that was designed for open water. The next larger watershed was the upper non-tidal half of the Patuxent watershed that drained to the USGS gage at Bowie (940 km²).

And finally we examined the whole Patuxent watershed (2,352 km²).

The results of surface water flow calibration have been reported elsewhere [Voinov et al., 1998; Voinov et al., 1999]. They were in fairly good agreement with the gage data and what is most important, we had enough control over the hydrologic processes to modify the patterns of the hydrographs in the way we needed. Spatial nutrient dynamics were calibrated for data at several gaging stations on the Patuxent.

In contrast to the water quality modeling, the plant module is less dependent upon spatial interactions. Therefore the unit model calibration in this case is more important. The calibrations were carried out for a series of parameter sets, representing the different habitat types and plant communities associated with them. We have considered a forest habitat, and a number of agricultural habitats, such as, corn, winter wheat, soybeans and fallow.

In this case it was most important to make sure that the available parameter set is sufficient to represent a variety of plant behaviors for different habitats and control factors. The overall pattern of growth, maturity and decay is similar for most of the plants, however the dynamics of photosynthetic and non-photosynthetic biomasses varies for different plants. In Fig. 3 we represent the various growth curves for different habitats. The module producing these curves was the same Plant Module; only the parameter sets were different.

Once the local modules were tested and precalibrated, they were translated and compiled by the SME into a general spatial model that has been then applied to the full Patuxent watershed and its subwatersheds. The resulting model has been then used to run an extensive scenario analysis program [Costanza, et al., 2002]. By running these scenarios we effectively further test the model for an even wider range of parameters and forcing functions, which considerably increases our confidence in its robustness and feasibility. For more details about PLM see our web site at <http://iee.umces.edu/PLM>.

Another recent application is on the Imandra Lake watershed, Kola Peninsula, Russia. In this case the modular approach is essential to represent a wide variety of ecological and socio-economic processes (see <http://iee.umces.edu/AV/KolaModels>).

5. CONCLUSIONS

In the modular approach we do not intend to design a unique model that is to be further applied to a variety of localities. In this case our goal is to offer a framework that can be easily extended and

Figure 3. Plant dynamics in various habitats. Whereas agricultural habitats (B, C, D) are harvested, the forest (A) keeps accumulating the biomass of non-photosynthetic tissue. Note that winter wheat (D) is displayed over a 2-year time interval.

is flexible to be modified. A module that performs best in one case may not be sufficient in another. The goals and scale of a particular study may require a completely different set of modules that will be invoked and further translated by the SME into a working model. Though STELLA may not be perfect for all models and processes involved, by using it to describe the modules we can provide the transparency that is essential for reimplementing modules in different contexts and environments.

For a researcher a model is predominantly a tool for understanding the system. By plugging together a number of black boxes, which specifics and behavior is obscure and hardly understood, we do not significantly increase our knowledge about the system. From the complex natural system we go to a complex computer system that may be also too hard to deal with. The results generated are difficult to interpret, when there is not enough understanding of the processes that are actually modeled. The decomposition of systems requires careful analysis of spatial and temporal scales of processes considered and is very closely related to specific goals of the model built.

In this context the modular approach can be useful for modeling, if the focus is shifted from reusability and "plug-and-play" approach to transparency, analysis and hierarchical description of various processes and system components. With the modules being transparent and open for experiment and analysis, the researcher can better understand the specifics of the model formalism that is inherited. It is easier to decide whether the modules provided are suitable or if they should be modified and tuned to the specifics of the goals of

the concrete study. It is mostly for this reason that we offer the modules in the LHEM in the STELLA implementation that provides the needed transparency and openness.

There have been several attempts of collecting, documenting, and archiving environmental models [Fiddman, URL; Noble and Davies, 1995; CEML, 1997], among which the Kassel University Register of Ecological Models is probably most noteworthy [Benz and Knorrnschild, 1997; Hoch et al., 1998]. However in our case we seek quite a different goal and instead of collecting all the available models and approaches we tend to limit our scope to a minimal set of modules needed to represent a fairly wide variety of environments. In this respect this approach can be compared to the MML system [Leavesley et al., 1996], where the goal is also to present certain modules and models in a format that would allow recombination and modification to meet the user's needs. This does not preclude expandability of the system. Additional modules are invited and can be added to the system, as long as they comply with a certain set of rules that would allow their integration into the system.

A module that is calibrated for one system or locality will most likely require recalibration when applied to a different case study. Moreover, a module calibrated as stand-alone will need further calibration when integrated into a more complex structure and running along with other modules. However, we argue that the calibration of individual modules and further recalibration of combinations of pre-calibrated modules is a simpler task to accomplish than to calibrate an immensely complex integrated model from

scratch. Besides if we provide means to store and document the intermediate parameter sets that were found optimal in various case studies, we can significantly simplify the reuse and recalibration of modules in the future.

When applying the LHEM, or any other modeling library, the major complication for the user is to put together the modules in a meaningful and consistent way. In any prefabricated model, the issues of scale consistency were predefined and presumably taken care of by the model developers previously. Now with the modular approach the challenge of integrating the modules in such a way that they match the complexity of the system at study and be mutually consistent becomes the task of the library user. Once again this added concern is the price that is paid for the added flexibility and optimality of the resulting models. In theory, we can envision modeling systems that would keep track of the scales and resolutions of the various processes involved, and automatically allow links with only such modules that would match these scales. In practice, with all the complexity and uncertainty associated with ecological and socio-economic systems, it may still be a while until such modeling tools will appear. In the meanwhile we think that the module transparency will be a very important prerequisite of modularity, especially if the modules are to be used in a research context.

6. ACKNOWLEDGEMENTS

Funding for this research has been provided by the EPA STAR program, Office of Research and Development, National Center for Environmental Research and Quality Assurance (R827169), and by the U.S. National Science Foundation, Office of Polar Program (OPP #0095196).

7. REFERENCES

- Benz, J. and M. Knorrenschild (1997). Call for a common model documentation etiquette. *Ecological Modelling* **97**(1,2): 141-143
- CEML (1997). Civil/Environmental Model Library. Department of Civil and Environmental Engineering, Old Dominion University, <http://www.cee.odu.edu/cee/model/model.html>
- CORBA (1996). CORBA Object Management Group. <http://www.omg.org>
- Costanza, R., A. Voinov, R. Boumans, T. Maxwell, F. Villa, H. Voinov and L. Wainger (2002). Integrated ecological economic modeling of the Patuxent river watershed, Maryland. *Ecological Monographs* (In press).
- Fiddman, T. System Dynamics Model Library. <http://home.earthlink.net/~tomfid/models/models.html>
- Fishwick, P.A., D.R.C. Hill and R. Smith, (Eds.) (1998). Proceedings of the 1998 International Conference on Web-Based Modeling and Simulation. San Diego, SCS.
- Hoch, R., T. Gabele and J. Benz (1998). Towards a standard for documentation of mathematical models in ecology. *Ecological Modelling* **113**: 3-12. <http://dino.wiz.uni-kassel.de/ecobas>
- HPS (1995). STELLA: High Performance Systems. <http://www.hps-inc.com/edu/stella/stella.htm>
- Leavesley, G.H., P.J. Restrepo, L.G. Stannard, L.A. Frankowski and A.M. Sautins (1996). MMS: A modeling framework for multidisciplinary research and operational applications. In: GIS and Environmental Modeling: Progress and Research Issues. Fort Collins, CO: 155.
- Macey, R.I. and G.F. Oster. (1993). Berkeley Madonna. <http://www.berkeleymadonna.com/index.html>
- Maxwell, T. (1999). A parsi-model approach to modular simulation. *Environmental Modelling and Software* **14**: 511-517
- Maxwell, T. and R. Costanza (1997). A language for modular spatio-temporal simulation. *Ecological modelling* **103**(2,3): 105-114
- Noble, I. and I. Davies (1995). MUSE - A Multi strata Spatially Explicit ecosystem modelling shell. <http://biology.anu.edu.au/research-groups/ecosys/muse/MUSE.HTM>
- Reynolds, J.F. and B. Acock (1997). Modularity and genericness in plant and ecosystem models. *Ecological Modelling* **94**(1): 7-16
- Sequeira, R.A., R.L. Olson and J.M. McKinion (1997). Implementing generic, object-oriented models in biology. *Ecological Modelling* **94**(1): 17-31
- Silvert, W. (1993). Object-oriented ecosystem modeling. *Ecological Modelling* **68**: 91-118
- Villa, F. and R. Costanza (2000). Design of multi-paradigm integrating modelling tools for ecological research. *Environmental Modelling and Software* **15**
- Voinov, A. and A. Akhremenkov (1990). Simulation modeling system for aquatic bodies. *Ecological Modeling* **52**: 181-205
- Voinov, A., R. Costanza, L. Wainger, R. Boumans, F. Villa, T. Maxwell and H. Voinov (1999). Patuxent landscape model: integrated ecological economic modeling of a watershed. *Journal of Ecosystem Modelling and Software* **14**: 473-491
- Voinov, A., C. Fitz and R. Costanza (1998). Surface water flow in landscape models: 1. Everglades case study. *Ecological Modelling* **108**(1-3): 131-144
- Voinov, A., H. Voinov and R. Costanza (1999). Surface water flow in landscape models: 2. Patuxent case study. *Ecological Modelling* **119**: 211-230