Wetlands Program

June 1992 No. 92-6

Technical Report

College of William and Mary
Virginia Institute of Marine Science
School of Marine Science
Wetlands Program
Gloucester Point, Virginia 23062

Program Director: Dr. Carl Hershner
Editors: Kirk J. Havens
Maryann Wohlgemuth

Commonwealth's Declared Policy:

"to preserve the wetlands and to prevent their despoliation and destruction. . ."

This report was funded by the Wetlands Program of the Virginia Institute of Marine Science.

Printed on recycled paper

The State of the Art In Wetlands Modelling: An Emphasis on Techniques and Technologies

William J. Seufzer

Introduction

Today's scientific body of knowledge continues to grow along with the number of questions science and society must answer. To exercise our understanding of this knowledge and provide at least partial solutions to questions, models are often constructed. A model can be a simple drawing showing interactions between components in a system or a very complicated mathematical description. Models can be static, like a drawing or equation, but can become dynamic when transposed into a computer simulation. While no model may ever be exact, modelling is an effective tool in testing and expanding the understanding of natural processes.

Like any tool, modelling and simulation evolve with time. In the recent past analog computers (electronic circuits designed to simulate mathematical equations) were the state of the art technique for simulation, today there are digital computers. While modelling has been used in aerospace, economics, and many other areas, this paper will focus on wetlands modelling.

State of the art in wetlands modelling has been summarized recently by Mitsch (Mitsch et al, 1988). Mitsch's book characterizes the state of the art as applied to various wetland types. While it is important to be aware of WHAT models have been built WHERE, it is equally import to concentrate on HOW these models are built.

This paper will summarize the current state of modelling and simulation in terms of techniques and technologies. Where it is possible examples will come from wetlands literature. The first section of this paper will describe the major steps required to build any simulation. The second section emphasizes techniques and technologies used in building simulations and where they fit within the six major steps.

(continued)

The Six Major Steps

The model building process can be summarized in six major steps. This breakdown was chosen as it fits nicely with describing some advanced tools and how they fit into the model building process. Other authors may chose to lump some of these steps together or to expand a step into several.

Definition

Before any model can be constructed the scope of its existence must be defined. There are no established techniques or technologies that assist definition, but the concept is straight forward. The model must have a purpose or goals and be placed in time and space (Grant, 1986). Model definition is perhaps the most important step as it sets the boundaries within which the model will exist. Models with different purposes and goals will have different structures and components. For example a model of hydrodynamic processes in a wetland will differ from one emphasizing succession of plant species. A model trying to determine the effects of hydrodynamics on plant succession is another possibility and may have a unique set of goals.

Some specific areas of wetlands modelling include: ecological processes, ecosystem dynamics, impact assessment, and management (Ondok, 1990; Jorgensen et al, 1990; Finn, 1990; Mitsch et al, 1990). These papers are mentioned here as examples of different model purpose and goals. For instance, while the purpose of a model may be management oriented there are several types of management models listed. Each model type has its own analysis goal such as, wetland removal, hydrologic modification, biotic harvesting,waste dispersal and recycling, and fish & wildlife protection.

The main point of definition is to establish model boundaries and context. Even with the same boundaries, models with different goals will take on a different context. To keep focus in the remaining steps of model building, a clear definition of what is to be modelled must be made.

Conceptualization

This step in the modelling process is just as critical as defining the model goals. Based on the established goals this step identifies and relates the important elements of the system. Sklar (Sklar et al, 1990) groups definition and conceptualization together and summarizes various methods for developing conceptual models. While there are several techniques, three of them are of interest here; Input-Output matrix, Forrester Diagrams, and Energy Circuit diagrams. These are all described in the previously mentioned Sklar text but carry some important implications.

An input-output matrix construction can be used in conjunction with the black-box , signed digraph, Forrester, or energy circuit methods for later use with network analysis (described later).

Forrester and energy circuit methods make use of graphic symbols to represent a model. The graphical language helps for quicker understanding of a model by using different symbols for storage elements, controls, material flows, etc. The technique of using a graphical language is very desirable as will be shown later.

Generating Equations

When all the important elements and their relationships are established the next step is to describe the relationships mathematically. Ondok (Ondok, 1990) has a collection of equations for a variety of ecological processes. These include energy transfer, adaptation, irradiance, and photosynthesis. Equations may also come from a researchers own work in which case a simulation may be built to test the newly derived equations.

There doesn't seem to be any high tech method for generating equations. They must be derived, estimated, or borrowed. As will be described later, the state of the art for equations may be to eliminate them entirely through the use of fuzzy rules in an expert system.

Computer Code

When all of the equations have been defined it is time to put them together in a meaningful way. At this point the model is in the process of becoming a simulation. If a programming language like Pascal or FORTRAN is used the equations must be transposed into that language. Furthermore, code must be built around those equations to keep track of time, print variables to a file for later analysis, and other bookkeeping functions. For a person without any advanced tools and who is not used to programming, this step can involve a large amount of time.

Once the code has been generated, it must be verified. This step involves testing the numerical methods used in the simulation to make sure they accurately reflect the equations. A check is also necessary to make sure round-off and truncation errors are within allowable limits. Verification is treated as a sub-step in this paper but can also be considered a step on its own.

Validation

Once a simulation is built and seems to be running properly a method must be developed to verify that its results coincide with the system being modelled. Statistical methods such as the *Student t* test might be performed to compare model results with collected data. Parrish (Parrish et al, 1990) suggests that comparing averages from a model to collected data may not be valid and proposes a quantitative method for testing whether or not model variables are within a prescribed factor of true values.

Sometimes historical data may be used to validate a simulation. In a model of coastal landscape dynamics, Costanza (Costanza et al, 1990) made use of US Fish & Wildlife Service data and published findings to calibrate a large distributed simulation of 2479 interconnected square cells. Each cell represented 1 square kilometer of the Atchafalaya delta. Historical data were available for 1956, which defined initial conditions, and for 1978 & 1983, which became correlation points. Once the simulation was tuned projections could be made to assess water flow changes and primary production due to man-made changes such as dikes and bridge pylons.

Analysis

In this phase the simulation output is looked at. If one of the design goals was to simulate the effect of sea-level rise on marsh succession, then varibles describing these effects within the simulation would be viewed. Analysis of these variables can lead to insight about the system or force you to reevaluate your design (simulations can be wrong).

In addition to watching simulation variables, analysis can be done on the relationships between the simulations components. Following from the input-output matrix conceptualization is an interesting analysis approach called network analysis. Christian (Christian et al, 1991) and Baird (Baird et al, 1989) have used this approach applied to nitrogen and carbon, respectively, in estuarine nutrient cycling. This analysis approach is based on network theory and provides a method to assess various direct and indirect pathways between organisms in the system. The analysis can be used to characterize ecosystem behavior as a whole system instead of a group of separate interactions.

Techniques for Model Construction

The previous section outlined steps to follow when building a simulation. The steps form a conceptual basis for WHAT must be done. Now techniques to use in carrying through with those steps will be outlined. Techniques deal with the HOW of simulation construction.

Brute Force

This technique is referred to as brute force because the modeler is forced into performing all the work in model building. All six steps are done with little computer assistance. The models definition, conceptualization, and equation definition have been accomplished on paper at this point. It is time to build the simulation.

The majority of work is in the step of writing and debugging computer code. In the brute force scenario every line of code is written by the modeler in a language such as Pascal, C, or

FORTRAN. All equations, connections, time control, output, and numeric methods code must be written and debugged. Assistance might come from using a library of routines (code that has already been written) that perform the numerical methods functions and help with driving output devices such as plotters.

For models with less than 10 components this may be a reasonable technique to use if no other tools are available. As simulation size increases in components and complexity this technique can become intractable. Even with good organization on the part of the programmer, adding a component or modifying a flow may require editing of code in several places. Without extreme care the model can become invalid and seem impossible to correct.

Simulation Languages

To help in the code generating phase, simulation languages have been developed. There are several of these types of tools commercially available. Basically these tools use their own language that is designed specifically for simulation building. In this approach much of the overhead code writing for numeric methods, time control, and output are taken care of in a few relatively simple steps.

Like using a programing language, these tools require a moderate level of computer science literacy. Nolan (1991) makes use of an Expert System interface to a general purpose simulation language. This combination presents an intelligent simulation environment where the user draws the model to the computer and code is automatically generated. The user can watch a run-time animation of the simulation and the package automates some the post-processing of results with links to word processors, plotting devices, and graphics packages. An application similar to this, and available at a low cost, is mentioned in the following section.

Special Applications

Applications such as STELLA (High Performance Systems, 1990) have now become available and help in several steps of the model building process. STELLA helps with the conceptualization stage by allowing the user to draw the system on the computer screen. The drawing language is similar to the Forrester and Energy Circuit languages. For an example use of STELLA see Mitsch (1988) and Figure 1 (pages 6 and 7).

As the modeler draws a system, STELLA keeps track of the components and their interconnecting flows. Equations must then be entered by the modeler to describe how the incoming flow affects the components state. Changes made to the conceptual model on the screen update STELLA's internal mathematical representation. The modeler does not have to write code at all, only the necessary equations.

Further, STELLA takes care of integration, timing, and has tools for output. Graphing tools allow the user to watch several variables as the simulation runs. Validation and analysis are left to the user but this is still a considerable step forward from the brute force technique.

Object-Oriented Programming

An object-oriented approach is a technique that at first might appear as a small evolutionary step up from the brute force technique. Equations must still be transposed into a computer language and numeric methods and data output must still be written or purchased. But an object oriented approach provides an organizational structure that could allow large simulations to be built and easily modified.

While it is beyond the scope of this paper to review the object-oriented philosophy, the technique shows its strength in individual-based simulations (a simulation where each fish can be considered individually instead of all fish appearing as one lumped state variable). Sequeira (1991) uses an object-oriented approach to simulate the growth of cotton. Individual organs of the plant are modelled to capture their behavior as a function of other plant organs. This object-oriented approach is a step towards simulating how individuals contribute to population patterns.

While an individual based simulation could be done in any language, the object-oriented approach provides a framework to make a detailed simulation easier to construct. If more complexity is needed in the simulation, objects can be added with, conceivably, less pain than as in the brute force technique.

Expert Systems

The use of expert systems in building simulations was introduced earlier. The expert system, given a set of rules, can assist a modeler in building a simulation and help in avoiding common pitfalls (error correction). It is conceivable that the expert could be used to catch errors like trying to represent carbon dioxide eating phytoplankton.

Another area of expert system application is in validation. Deslandres et al (1991) points out that validation requires a high level of knowledge. He proposes and demonstrates an expert system which assists the user in choosing an appropriate validation method, applying the method appropriately, and with interpreting the results.

Expert systems have been applied to landscape ecology in a very interesting way. Instead of using the system to aid in simulation development, it is used as a simulation tool. Sterbacek et al (1990) demonstrates the use of an expert system to make a prognoses on forest structure changes due to emissions and climate change. This approach eliminates mathematical equations and replaces them with fuzzy rules. The effects of 8 variables on a 9th are expressed as rules along with a weighting factor on the rule. The weighting factor expresses how certain (0-100%) the modeler is of the rule. Results from the system have been unthought of, yet acceptable.

Artificial Life

While expert systems are full of intrigue there is another technology being explored called Artificial Life, or a-life. In an article by Stein (1991) it is stated that biologists study living organisms from the top-down. A-life provides a framework in which to study from down-up, to see what could be. An example of an ant colony is used to demonstrate the concept. In short, specialized behavior evolves from originally random events all trying to attain the same goal. In successive generations of the artificial organisms, the randomness is out-bread as only the most successful organisms are allowed to reproduce.

Use of this concept is being applied to robot control (Freedman, 1991). This article is about Artificial Intelligence research being carried out at MIT by Rodney Brooks. Multi-legged robots are given a goal to reach, such as walk, with no instruction. Sensors on the robot indicate successful behavior. Initially a robot will randomly twitch and flounder until a successful control combination is found that achieves the goal of walking. "Intelligent" behavior seems to evolve with a minimum of programming and a minimum of computer power. The small robots have what they need on-board. This is in contrast to systems where large, remote computers are needed to control robots where the robot is given a complex model of a "human's" world.

Conclusion

This paper reviews the major steps of building a simulation model as a structure to review the state of the art in wetlands modelling. While examples were brought mostly from wetlands and biological literature, examples from computer science literature were brought in to show that simulation technologies continue to be born and evolve.

Currently, most simulations are still built by the brute force method, probably due to the low monetary investment required. But this low cost approach limits model size and complexity, complexity which needs to be modeled if models are to approach reality. As the advanced tools become cheaper and easier to use, ecologists will be able to build larger, more complex models.

It may never be possible for a model to be exact, but that isn't the point. The point is that models are an effective tool for organizing research and testing hypotheses. In creating a model we quickly have to confront what we don't know. Defining what we don't know is progress.

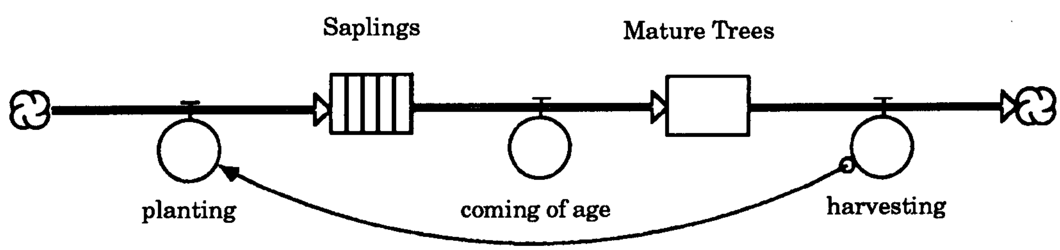
Figure 1. Building a simulation in STELLA® based on an example from the STELLA Users Manual.

Step One - Definition

A stand of 5 mature trees and 10 saplings. One Harvested tree is replaced by one new sapling. Natural natality and mortality are not considered. Many other factors such as climate are not considered. The simulation will show the effect of increased harvesting on the standing stock of mature trees.

Step Two - Conceptualization

The diagram illustrates Saplings and Mature Trees. Planting creates saplings. As saplings come of age, they are moved to the standing stock of mature trees. Harvesting of a mature tree causes another sapling to be planted. (The cloud symbols represent the trees coming from and going to anywhere. For our definition, this doesn't matter.)



Step Three - Generating Equations

Within the Stella Tool each component must be given its mathematical meaning. The equations below have been copied directly from STELLA's equation sheet.

```
Mature_Trees(t) = Mature_Trees(t - dt) + (coming_of_age - harvesting) * dt
INIT Mature_Trees = 5

INFLOWS:
coming_of_age = CONVEYOR OUTFLOW
OUTFLOWS:
harvesting = 10 + STEP(2,4)
Saplings(t) = Saplings(t - dt) + planting - coming_of_age) * dt
INIT Saplings = 10
   TRANSIT TIME = 1

INFLOWS:
planting = harvesting
OUTFLOWS:
coming_of_age = CONVEYOR OUTFLOW
```

Figure 1 (continued)

Step Four - Computer Code

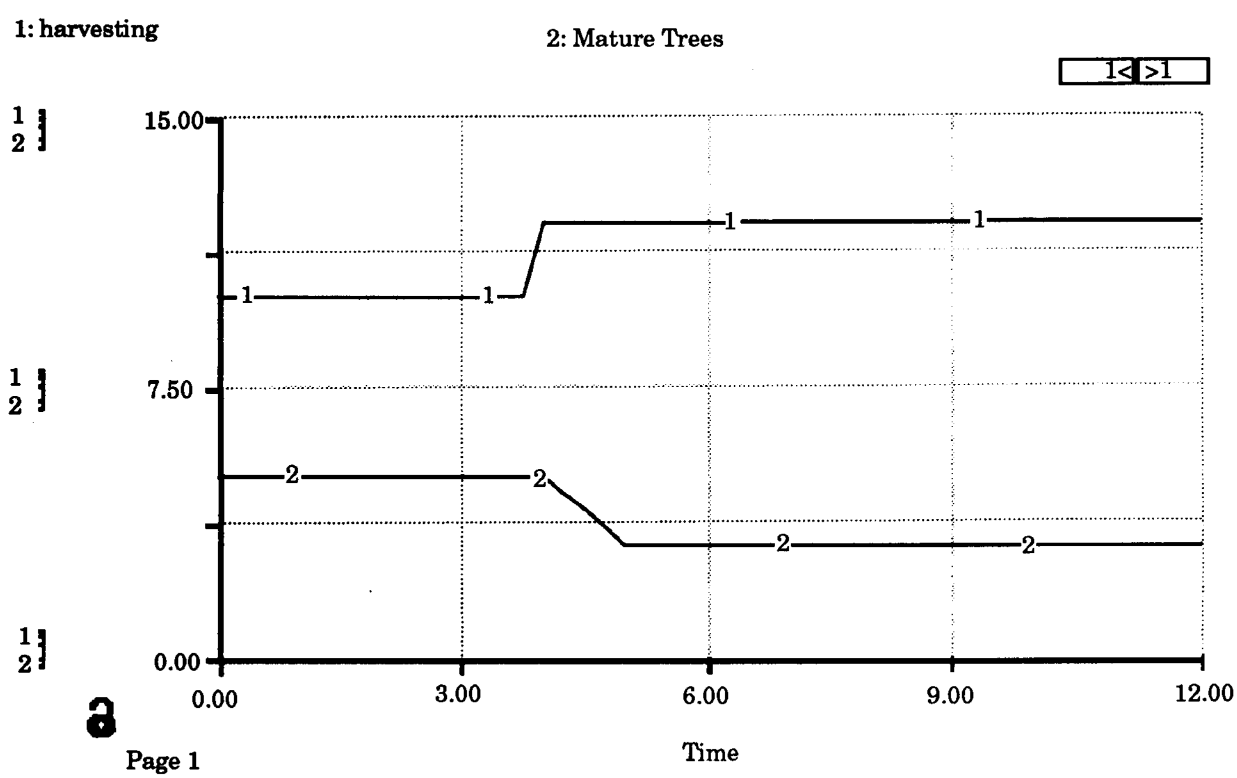
STELLA has done this for us. While we did have to go through the effort of entering the equations, all the routines for numeric methods, data storage, and plotting are taken care of by STELLA.

Step Five - Validation

This simulation has not been validated. But in a simple case, such as this, it is easy to see that the model is performing as designed. (Validation is left as an exercise for the student.)

Step Six - Analysis

The plot below shows harvesting and number of mature trees as functions of time. Note that as an additional 2 trees are harvested (12 total) in each time step, the standing stock of trees drops by 2. Does this seem logical?



STELLA is a registered trademark of High Performance Systems, Hanover, NH.

References

Baird, D. and Ulanowicz, R., 1989, The Seasonal Dynamics of the Chesapeake Bay Ecosystem, Ecological Monographs, 59(4).

Christian, R.R., Boyer, J.N., Stanley, D.W., and Rizzo, W.M., 1991, Network Analysis of Nitrogen Cycling in an Estuary, Paper submitted for: Modeling the Metabolic and Physiologic Activities of Microorganisms, C. Hurst (Editor), Wiley Press.

Costanza, R., Sklar, F., and White, M., 1990, Modelling Coastal Landscape Dynamics, BioScience, 40:2.

Deslandres, V., and Pierreval, H., 1991, An expert system prototype assisting the statistical validation of simulation models, Simulation, 56:2, 79-89.

Finn, J.T., 1990, Impact Assessment Models, Wetlands and Shallow continental Water Bodies, Vol 1, pp 703-725, Edited by B.C. Patten et al., SPB Academic Publishing bv.

Freedman, D.H., 1991, Invasion of the Insect Robots, Discovery Magazine, March 1991, pp 42-50.

Grant, W., 1986, Systems Analysis and Simulation in Wildlife and Fisheries Science, Wiley & Sons.

High Perfomance Systems, 1990, Stella II User's Guide, Hanover, NH.

Jorgensen, S.E., and Koryavov, P.P., 1990, Modelling Ecosystem Dynamics, Wetlands and Shallow Continental Water Bodies, Vol 1, pp 691-702, Edited by B.C. Patten et al., SPB Academic Publishing bv.

Mitsch, W.J., and Jorgensen, S.V., 1990, Modeling and Management, Wetlands and Shallow Continental Water Bodies, Vol 1, pp 727-744, Edited by B.C. Patten et al., SPB Academic Publishing bv.

Mitsch, W.J., 1988, Productivity-Hydrogology-Nutrient Models of Forested Wetlands, Wetland Modelling, Developments in Wetland Modelling,12, pp 115-132, Edited by WJ Mitsch et al, Elsevier Science Publishers B.V.

Mitsch, W.J., Straskraba, M., and Jorgensen, S. E. (Editors), 1988, Wetland Modelling, Developments in Wetland Modelling,12, Elsevier Science Publishers B.V.

Nolan, P.J., Lane, G. M., and Fegan, J. M., 1991, ISI - An environment for the engineering use of general purpose simulation languages, Simulation 56:1, 41-47.

Ondok, J.P., 1990, Modeling Ecological Processes, Wetlands and Shallow Continental Water Bodies, Vol 1, pp 659-689, Edited by B.C. Patten et al., SPB Academic Publishing bv.

Parrish, R., and Smith, C., 1990, A Method For Testing Whether Model Predictions Fall Within A Prescribed Factor Of True Values, With An Application To Pesticide Leaching, Ecological Modelling, 51: 59-72.

Sequeira, R.A., Sharpe, P.J.H., Stone, N.D., El-Zik, K.M., and Makela, M.E., 1991, Object-oriented simulation: plant growth and discrete organ to organ interactions, Ecological Modelling, 58: 55-89.

Sklar, F.H., Costanza, R., and Day, J.W. Jr., 1990, Model Conceptualization, Wetlands and Shallow Continental Water Bodies, Vol 1, pp 625-658, Edited by B.C. Patten et al., SPB Academic Publishing bv.

Stein, R.M., 1991, Real Artificial Life, Byte Magazine, January 1991, pp 289-298.

Wetlands Program ***Technical Report***

College of William and Mary
Virginia Institute of Marine Science
School of Marine Science
Gloucester Point, Virginia 23062 U.S.A.

NON PROFIT ORGANIZATION
U.S. POSTAGE PAID
GLOU. PT., VA 23062
PERMIT #6