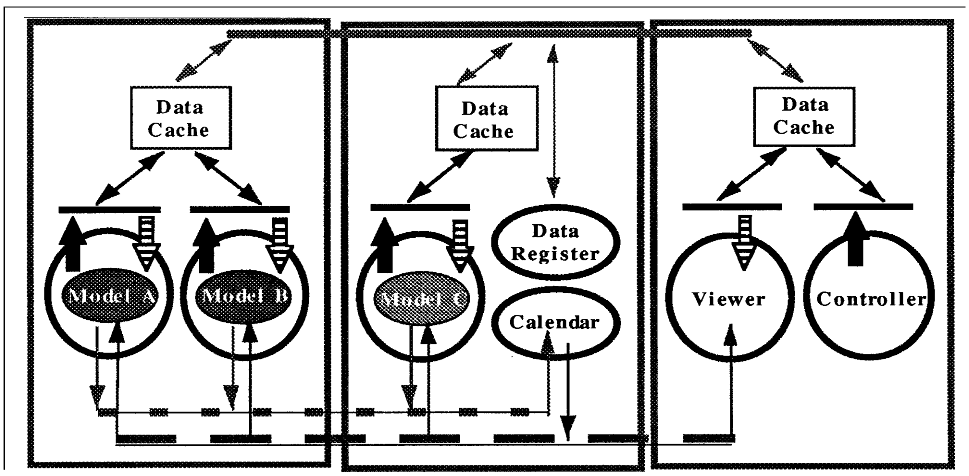
US Army Corps
of Engineers
Construction Engineering
Research Laboratory

USACERL Technical Report 98/78
July 1998

Integrated Spatio-Temporal Ecological Modeling System

Concept Design

by James D. Westervelt



Land management has always been a multidisciplinary effort that relies on the integration of discipline-specific simulation models. These models have traditionally been conceptual. Each model has been based on formal training, education, and experience with the landscape. Recently, many of the academic disciplines associated with land management have been capturing their conceptual models in formal computer models and simulations. Once formalized, such models may be more thoroughly studied, analyzed, and improved.

Science requires that the number of simultaneously changing variables in a model be held to a minimum so those that are allowed to change can be more thoroughly understood. While this allows a scientist to better understand a neatly defined aspect of nature, the resulting models are less than desired for the purposes of land management. Land management requires that intelligent decisions be based on projecting the state of the landscape under different management options, thus taking into account all salient aspects of the landscape. This document argues for the design and development of management oriented simulation models, that generically integrate a number of different domain-specific models.

Foreword

This study was conducted for the Directorate of Research and Development, Headquarters, U.S. Army Corps of Engineers (HQUSACE) under Project 4A161102BT25, "Environmental Research—Corps of Engineers"; Work Unit J07, "Fundamentals in Ecological Modeling." The technical monitor was Dr. Thomas Hart, CERD-M.

The work was performed by the Natural Resources Assessment and Management Division (LL-N) of the Land Management Laboratory (LL), U.S. Army Construction Engineering Research Laboratories (USACERL). The USACERL principal investigator was James D. Westervelt. Dr. William D. Severinghaus is Operations Chief, CECER-LL. The USACERL technical editor was Gloria J. Wienke, Technical Resources.

COL James A. Walter is Commander of USACERL and Dr. Michael J. O'Connor is Director.

Contents

1 Introduction

Background

Predicting the future has always been a dream of people. We naturally develop ideas about the future and make decisions based on our understandings about how these decisions affect our future situations. These ideas are, in fact, simulation models. They combine understandings of the current state of affairs with ideas about cause-effect relationships and are used to predict how the future might respond to alternative decisions. Because of the human preoccupation with the future, we are naturally employing the computer (hardware and software) to look into the future.

Today an array of computer-based simulations has been developed to assist Army land managers. These include plant and animal population dynamics, physiological mechanisms, overland water flow, stream and river flow, vegetation (natural and agricultural) growth, weather, climate, and plant succession. Now that such models have reached a level of acceptance, it is becoming increasingly difficult for managers to work with the output of two or more such models. Typically, each model is only able to simulate some fraction of all of the salient processes at some preset spatial and temporal scale. The output from several models, each running at different scales and focusing on different aspects of the whole, cannot easily be combined. A current challenge then is to integrate several different models in a manner that allows them to run against a single simulation clock and be able to exchange and share information. Such an integrated system will help Army land managers to manage the future of landscapes on installations.

Objective

The objective of this study was to create a design document that will facilitate construction of the fundamental tools and capabilities necessary to support the future simulation tools that will be used by Army land managers to manage military landscapes.

Approach

The first step in facilitating the construction of an ecological modeling system was to review the types of models that have already been developed. The next step was to define and formally capture the multi-disciplinary decisionmaking processes involved in land management. The third step was to integrate information from the first two steps and suggest a process for developing a Integrated Spatio-Temporal Ecological Modeling System (I-STEMS).

2 Modeling Background

For this discussion, models and simulations can be assigned to one of two basic types: scientific and management (or decisionmaking). Science is an activity that necessarily focuses on some relatively narrow aspect of nature. It seeks precision, understanding, and the development of general theories (models). Management is necessarily a very messy process compared to science. A scientific investigation can require years to understand a single aspect of nature, but management must make relatively quick decisions that may affect many aspects of nature at multiple scales in time and space while simultaneously affecting the lives and fortunes of people. Scientific investigations are typically funded in relatively small increments while management can easily be associated with decisions affecting entire cities and/or regions with significant economic impacts. The scientist says that we must not be hasty in making decisions; but the manager must not delay. This report focuses on improving the design, development, and application of management models.

Modeling for Scientific Understanding

C. Wissel (1992) identifies scientific models as being either descriptive or simulation. Descriptive models typically result from the statistical reduction of copious field data into simple relationships between salient features. These models are useful correlations with demonstrated statistical significance. However, they provide no insight into how the system works. That is, correlation does not provide a cause-effect understanding. Simulation models capture cause-effect relationships and express fundamental descriptions of how nature works. Wissel argues that scientific models cannot become so complicated that they lose their explanatory potential. With too much complexity it becomes humanly difficult or impossible to understand what is happening in the model. We cannot then draw scientific generalizations about the system. Starfield and Bleloch (1986) agree and developed a list of problems and limitations associated with using very complex models for scientific investigation:

- Idealizations and abstractions are inherent properties of (scientific) models. Therefore, one cannot include all ecological factors and details in a model. A criterion for selecting factors must be given. Not specifying the aim of a model, but calling it a realistic approach, is an unrealistic pretension.

- If one really models an ecosystem in detail, it is difficult to draw general conclusions. On a very detailed level, all ecosystems differ from one another. But the aim of science is general statements and not the description of a singular case.

- Models with a high number of adjustable parameters can be fitted to almost everything. They do not have much predictive or explanatory power.

- Normally these complex models cannot be sufficiently presented and explained because all the assumptions and details cannot be given in a report of limited length. Therefore, they are not open to critical analysis and consequently are of little scientific use.

- The most important shortcoming of these models is that they are notoriously unsuitable for obtaining an understanding. It is impossible to say which of the many details of the model are essential for a particular result.

Scientific journal articles and books resulting from basic research efforts are necessarily simple, make a number of significant assumptions about the system being studied, and seek to have general application. Scientific investigations at the landscape and ecosystem level find it very difficult to identify general laws or truths that hold across a large number of sites and locations. Most studies result in descriptions and reductions of data collection exercises. General laws in ecology are few and far between, owing to the diversity in life's adaptation to different inorganic and energy conditions.

Some models have been developed for the scientific exploration of landscape dynamics. Energy input/output matrix modeling was developed for characterizing landscape processes (Hannon 1973; Hannon 1985). These models failed to capture the life interactions between organisms (such as disease, pollination, propagation, and animal behavior) and were less successful at capturing the essence of what takes place in a natural system.

To capture the dynamics of spatial distribution of resources H. Caswell developed the notion of "neutral models." These are simplified computer landscapes that are randomly generated to provide patterns of suitable and unsuitable habitats (Caswell 1976; Gardner, Milne, et al. 1987; O'Neill, Gardner, et al. 1992). For example, assume an animal or plant is constrained to live only in the suitable habitat and has no possibility of even crossing unsuitable areas. A number of questions can be posed to such a system. Gardner, Turner, and associates (1991) demonstrated that below a landscape coverage of 0.6 (60 percent) patches are highly fragmented. Their simulations demonstrated "... that large differences in species abundance and habitat utilization are produced by small changes in the maximum possible dispersal distance." Turner, Gardner, and associates (1989) used percolation models to evaluate disturbance intensity and frequency on various densities of habitat in neutral maps. Disturbance frequency and intensity had variable impacts on neutral model landscapes. When the landscape was occupied by less than about 50 percent of the habitat, that habitat was sensitive to disturbance frequency, but demonstrated little difference in its response to disturbance intensity. Habitats occupying more than 60 percent of the landscape were less sensitive to disturbance frequency, but more sensitive to disturbance intensity. O'Neill, Gardner, and associates (1992) through random models, showed that hierarchically structured landscapes (vs. random neutral model landscapes) had smaller perimeters, were less clumped on sparse landscapes, and were more clumped on dense ones. This permits percolation on a broader range of conditions.

The theoretical underpinnings of the need to explicitly acknowledge and model the spatial arrangement of resources are found in the theories of island biogeography (MacArthur and Wilson 1967) and early arguments supporting notions of metapopulation theory (Andrewartha and Birch 1954). Island biogeography provided a theoretical and mathematical framework for describing the relationships between a set of stable populations and areas of unstable populations (islands). Metapopulation theory allows interactions between numerous areas containing unstable populations. It provided a theoretical foundation describing processes by which similar competitors can coexist in a patchy environment (Levins and Culver 1971; Horn and Mac Arthur 1972; Slatkin 1974). J. Wu, J.L. Vankat, and associates (1993) studied patch dynamics as a function of connectivity, density, and arrangement. They found that minimum viable populations (MVP) must exist in at least one patch for the populations to persist. For guaranteed persistence of the population, a higher critical size is required. These metapopulation models presume that the populations exist in an environment with patches of resources that are adequate for the salient species sitting in a sea of inhospitable conditions. Individuals of the species must actively or passively cross the inhospitable areas to colonize the suitable patches.

Simple, but powerful, spatially explicit numerical models developed by R. Levins formed the foundation for a body of literature exploring metapopulation theory and its relationship to metacommunities, landscape ecology, island biogeography, patchy environments, and conservation biology. For a review, see Hanski and Gilpin (1991). Metapopulation theory provides a simple mechanism that explains how it is possible for a landscape to contain a number of direct competitors. In a completely homogeneous environment, the most successful competitors crowd out their inferior competition. Real systems are patchy at all levels of hierarchical organization because of perturbations and disturbances. In such dynamically heterogeneous environments, metapopulation theory predicts the existence of a potentially unlimited number of close competitors. Levins' basic equations have been extended in various different ways. Hanski (1985) added migration to Levins' model (to create a 3-state model). Dynamic complications, caused by immigration, were demonstrated to result in alternative stable equilibrium. Gilpin (1990) demonstrated numerical computer models for making predictions of the dynamics of real systems using metapopulation theory. Finally, Gardner, O'Neill, and associates (1993) conducted theoretical simulations of competing species with varying perturbance regimes and harvest schemes. The Levins model has also been applied to the European Badger. A Markov chain model was developed to represent the populations. Explorations of the model found that under certain circumstances it fit field data on the Badger (Verboom and Lankester 1991).

Metapopulation theory, island biogeography theory, energy input-output models, the Levins model, and percolation theory all provide some insight into the workings of landscapes and point to some notions of proper management. The goal for simplicity and general application leaves all of these models less than satisfactory for understanding and modeling any particular landscape. More recently a fuzzier theory has moved into the forefront of ecological scientific theory in response to the need to be applicable to a broader range of ecological settings — the hierarchy theory.

Hierarchy Theory

The predictability of ecological systems is inherently limited and is dependent on the scales (May 1986; Levin 1989; Vasconcelos, Ziegler, et al. 1993; Klijn and Udo de Haes 1994). The degree to which any given ecological study identifies the existence or non-existence of processes that allow a perturbed system to return to some equilibrium state depends on the temporal and spatial scales and the level of organization on which the study focuses. “Therefore, there is no single correct scale of investigation and thus no universal law in ecology” (Wu and Loucks 1991).

Wiens, Addicott, and associates (1985) write “Some of the most vociferous disagreements among ecologists arise from differences in their choice of scale.” To illustrate the point, they suggest how differently ecologists studying the relationships between jackrabbits and coyotes at five different scales might view their interactions. These scales were defined as: (1) the location where the entity lives, (2) a local patch occupied by many individuals, (3) many local populations that interrelate through dispersal, (4) a closed system (or approximation thereof), and (5) a biogeographical scale where different climates and different sets of species exist. Depending on the spatio-temporal scale chosen, two species can appear to be highly interrelated or completely independent. Land managers, modelers, and ecologists must always be willing to back away from a particular model or approach and view the system from perspectives arising from different scales in time and space. This will ensure that the proper scale is chosen with respect to the particular question or set of questions being asked.

Hierarchy theory offers a framework within which to view and integrate different scales. The theory has matured sufficiently to be documented in several books (Allen and Starr 1982; O’Neill, Johnson, et al. 1989). There are three dimensions: time, space, and organization. Organization refers to organizational levels of life, which are often viewed as nested. Atoms are organized into molecules, molecules into cells, cells into organs, organs into individuals, individuals into populations, populations into communities, and communities into ecosystems. Natural phenomena, which are represented by a large number of samples at the scale of study (e.g., atoms of an element in a sample or mice in a county) can be handled very well through statistical approaches and are called large number systems. Phenomena that are represented by very few samples can be handled by careful and thorough study of each sample (low number systems). Landscapes, when studied at the human scale, have too few components to treat statistically and too many components to study each thoroughly. Such “middle number” systems are the focus of hierarchy theory (Allen and Starr 1982).

Hierarchy theory links the levels of time, space, and organization. Lower organizational levels operate in smaller partitions of space and shorter periods of time. Individuals operate on small scales in time and space, while ecosystems operate on much larger scales in time and space. The apparently neat relationship between these three scales has been discussed and graphically depicted in time-space diagrams. Ocean hydrodynamics (Stommel 1963) and processes in landscape ecology (Urban, O'Neill, et al. 1987) have been presented in such diagrams showing a clear and simple relationship (Johnson 1993). Delcourt and Delcourt (1988) partition time and space into four domains (overall time and space of interest):

- *Micro-scale (1-500 yr, 1-10^6 m^2):* This domain is the most familiar to ecologists. Within it exist population dynamics, productivity, competition, and response to disturbance events.

- *Meso-scale (10^4 yr, 10^{10} m^2):* Here landscape mosaics and watersheds dominate. Animals and plants develop adaptation to disturbance regimes.

- *Macro-scale (10^6 yr, 10^{12} m^2):* This scale involves quaternary studies. Species displacements occur on a subcontinental scale, and rates of spread of species and genetics as well as extinctions define this scale.

- *Mega-scale ($>10^6$ yr, $>10^{12}$ m^2):* Planetary phenomena like development of biosphere, lithosphere, hydrosphere, and atmosphere, and macro evolutionary history of life on earth dominate at this scale.

According to hierarchy theory, systems result from evolutionary processes that favor a nested, hierarchical organization. Each level is constructed from identifiable subsystems (Johnson 1993). Hierarchical levels are separated by conceptual surfaces. For ecological modeling, the modeler need normally consider only three levels: (1) the level dealing with the question being asked of the system, (2) the next higher level to provide context (constraints), and (3) the next lower level, which contains the dynamics and structure to be modeled (Johnson 1993). Dynamics of even lower level structures in the hierarchy are, for the most part, sufficiently attenuated to be replaced by average behaviors or even ignored because they are captured in an attenuated and aggregated fashion through dynamics occurring in intermediate levels (O'Neill, DeAngelis, et al. 1986). Landscape ecologists, for example, attempt to capture the complexities at smaller-than-landscape scales into single numbers and indices (Turner 1989).

Hierarchy theory is clearly viewed by mainstream ecologists as an inescapable philosophy. Urban, O'Neill, and associates (1987) advocate a hierarchical paradigm to better understand the patterns in landscape ecology. Vasconcelos, Zeigler, and associates (1993) insist that multiple levels must be studied simultaneously. Wu says that the "hierarchical patch dynamics paradigm" is emerging, adding, “We must focus efforts on both process and context as well as the multiplicity of their temporal, spatial, and organizational scales” (Wu 1992). This certainly makes it difficult to conduct scientific experiments in ecology as prescribed by Wissel (1992) which are simple and easy to understand.

Land Management Modeling

The management of landscapes requires the combined application of scientific models developed by a variety of academic disciplines including economics, range management, ecology, forestry, regional planning, landscape ecology, hydrology, and politics. Hierarchy theory, though messy for the purposes of scientific investigation, is clearly useful for land management. Not only must knowledge from a variety of disciplines be considered, but the information is associated with different hierarchical levels of time, space, and natural organization. From the perspective of a scientist, the gap in knowledge required for adequately managing nature is vast and should include at least some human intervention. However, the fact is that the human population is intimately intertwined with nature, and impacts on nature are inevitable and, arguably, natural. Decisions will be made regardless of the gap in knowledge required to make the best, the optimal, or perhaps even wise decisions. Decisions that affect natural systems must be made and must be made with the best available science.

Models

Models developed to support land management are typically conceptual models that we hold in our conscious (and subconscious) minds. Chapter 3 explores how this approach is being augmented with the more formal capture of models in computer software. In this Chapter we review the types of models that have already been developed explicitly for management purposes.

Because management involves not only the best use of natural resources, but also the best use of capital resources, the development of natural resource computer models must be judged to be economically cost effective. With the application of any new technology, the costs are high. Therefore, only those projects with the biggest foreseen benefit will dare try out the new technology. Forest management models fit this bill in the 1970s as they are agricultural models associated with very expensive forest harvesting operations (Loucks, Doyle, et al.; Botkin, Janak, et al. 1972; Botkin, 1977; Daniels and Burkhart

1988; Fulton 1991). Another type of situation with very large potential benefits from modeling are political disputes over the use of large tracts of land for alternative purposes. An example of this is the design and development of a spatial simulation of a wetland that contrasted the benefits of using a coastal wetland for fishing, oil exploration and extraction, and recreational purposes (Sklar, Costanza, et al. 1985).

Geographic information systems (GIS) became popular land management tools in the 1980s. The standard GIS environment, however, recognizes only half of the information required to predict the state of a future landscape. The standard GIS data base captures digital maps that describe the state of the system as measured in the past. The missing component is information about how the various components of a landscape interact with one another. When a system contains the current system state and rules describing interactions between landscape components, it becomes possible to project alternative futures.

A good number of examples exist that demonstrate the power and potential of combining state and dynamics. An ecosystem management model has been developed for the Canadian Parks Service (Buckley, Coughenour, et al. 1993) and applied to Elk Island National Park (central Alberta). Economics, forest growth, and animal population species components of a system have been integrated by Liu (1993) in a system called ECOLECON. The movement of sand dunes across a landscape captured in a raster GIS has been accomplished (de Castro 1995). Combining a raster GIS (GRASS), an expert system engine (CLIPS), and specialized C software resulted in a dynamic landscape that simulated vegetation cluster development between 1941 and 1990 in a Texas Savanna Landscape (Loh and Hsieh 1995). Vieux and Westervelt (1992) provided an example of linking overland water flow equations to a GIS model to predict runoff velocities and depths across an entire landscape. Another multi-scale spatial ecological modeling system is hierarchical and includes individuals, patch, and the whole landscape (Perestrello de Vasconcelos, Zeigler, et al. 1993). These are but a few of a growing number of landscape simulation modeling examples that have been accomplished at research institutions. Enough of these types of simulation have been developed that a growing number of modeling environments have also been developed in support of landscape management.

Modeling Environments Developed for Management

Ball and Gimblett have written “More complex models need to be developed to more thoroughly evaluate, monitor and simulate the functioning of ecosystems”

(Ball and Gimblett 1992). Based on their experiences with landscape simulations they identified two main problems. First, it is important to be able to run different parts of a simulation at different time and space resolutions. Second, multiple models need to run simultaneously. To address these needs, they described a system that could meet dynamic landscape simulation modeling requirements. It is called the Spatial Dynamic Emergent Hierarchies Simulation and Assessment System (SDEHSAS). This design combines a GIS base with a data base management system (DBMS) for sharing between different model components and a combination of neural nets and genetic algorithms to filter data requests (resulting in self-adapting models).

A general-purpose landscape simulation modeling environment called the Modular Modeling System (MMS) has been recently released into the public domain (Leavesley 1996). MMS combines GIS with a library of landscape simulation models that can be linked and parameterized as required for a particular landscape. SIMPLEX II is another ecological simulation environment (Wittmann 1994). Its authors advocate hierarchical programming; multiple levels of system organization should be developed for a complete and useful simulation model. Developers working at a gross resolution level see only major model components and the data flows established between them. Development at higher levels of organizational resolution yields more complexity within the individual components. This approach is also captured in the commercial modeling software called Extend.* Developers can create detailed models of system components at one level of resolution, and can then work with those components as distinct entities at another level of system development resolution. The Spatial Modeling Environment (Maxwell and Costanza 1993; Maxwell 1995) is another powerful simulation environment. It is raster based and allows model developers to create and apply simulation models simultaneously to each grid cell in a raster representation of a landscape. Finally, the battlefield simulation world has developed simulation models at a large number of government and commercial research institutions that must be linked to create complete training simulation models. Recently the Defense Modeling and Simulation Office (DSMO) has created the High Level Architecture (HLA), which specifies how these various simulations will be

*Imagine That, Inc., 6830 Via Del Oro, Suite 230, San Jose, CA 95119; voice: 408-365-0305; Fax: 408-629-1251; e-mail extend@imaginethatinc.com. Citing this commercial product is not intended to be an endorsement or recommendation by the U.S. Government.

required to communicate with one another (DMSO 1996). Some of these different modeling environments and standards are discussed more thoroughly in Chapter 8.

3 Changing Use of Models in Land Management

Computer-based modeling and, more recently, simulation modeling is being used to support land management. Some people advocate modeling; others believe modeling is little more than a waste of time. Models should be reliable, but most cannot be judged reliable because it is unfeasible to completely explore the parameter space (Denning 1990). Modeling, however, is perhaps inescapable.

All perceptions are models. Modeling and simulation is a basic human activity and is used to make sense of the world around us. We formulate understandings or models of the people, interrelationships, and physical entities around us. Intuitive experience-based models are automatically applied to the information that flows into our senses to rapidly comprehend what we are currently experiencing. Information entering the senses is sampled for salient features which are then matched to potential models of what might be in the sensory field. The consciousness then becomes aware of the selected model. This process works so well and so efficiently that the consciousness rarely studies the details of objects, ideas, or relationships in the outside world. Occasionally the system fails in adequately matching sensory inputs to the correct model. For example, a piece of rope on the ground might be mistaken for a snake. This is especially true if the viewer is “looking” for snakes out of fear. In such a situation the sensory inputs actually processed might include little more than the facts that there is a long, sinuous object that is brown in color. The mind matches all of these characteristics to the model of a snake and returns the image of the snake model to the consciousness. The consciousness then sees not only the characteristics actually present, but also other characteristics like the pattern of scales, eyes, and flashing tongue. Similarly, we have our models of other people. Over time and experience with an individual, the model of that person becomes increasingly complete. We have models of standard individuals, situations, homes, cities, and ourselves. Our unconscious mind is continually seeking to match sensory input with these models. The point is that models are necessary and important components of human perception and understanding.

Although models certainly are commonplace, they provide some challenges in the workplace. Figure 1 depicts how a complex land management decision might be made in an office today. Each person involved in the decision possesses one or more conceptual models of how the landscape works. These are based on such things as formal academic training, experience, and one's innate ability to work with abstract concepts and models. Typically a number of individuals are involved. Each has a unique background that results in models more or less detailed in the areas of species requirements, habitat suitability, hydrology, biodiversity, genetics, chemical and noise impacts, etc. A question posed to the team is actually posed to the conceptual models resident in these individuals resulting in "best professional judgments." A political process typically resolves the conflicts between the different answers.

This approach works well and has been the mainstay of complex multi-disciplinary decisionmaking. There are some desirable improvements that seem to be possible with emerging technologies. First, the individual models are necessarily incomplete. Each individual is in possession of only a small part of the full breadth of information that is associated with the problem. Therefore, no one person typically possesses sufficient knowledge about the problem to be able to provide an optimal answer. This is why typically a number of individuals are involved. Each person represents a different academic background and field of experience. Second, the models themselves cannot easily be evaluated.

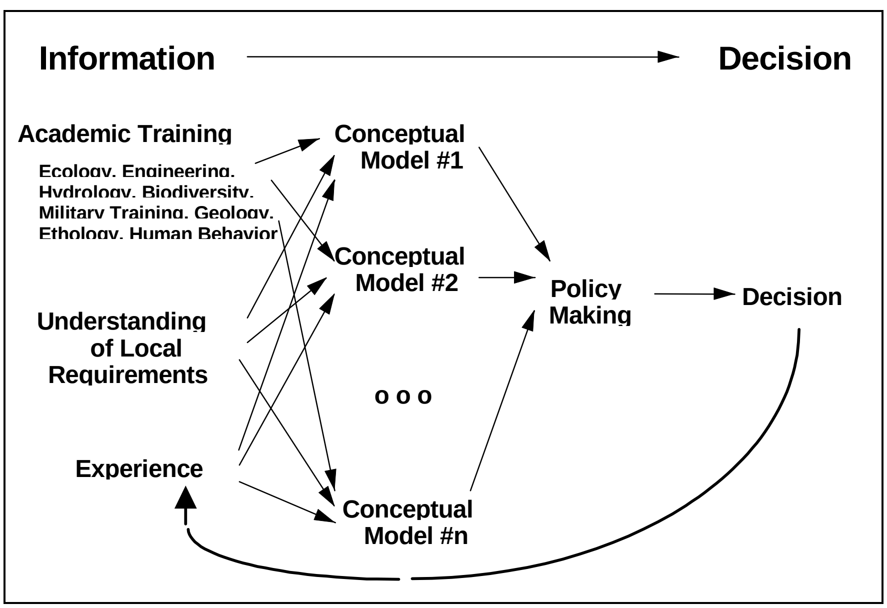


Figure 1. Current approach to landscape modeling.

Typically, each individual is not fully aware of the complexity of the models they are using, if indeed they are even aware they are using a model. Conceptual models are not, in practice, clean logical inference models, but rather pattern-matching models with the ability to interpolate crudely between patterns. Culturally, we speak with logic, but in our minds we think with patterns. To communicate efficiently, we must translate between the pattern-matching processes of thought and the formal logic of speech — a very difficult task. Only after the models are communicated can they be evaluated. Third, since models cannot be easily communicated, combining models is even more difficult and time consuming.

Should these conceptual models be formalized in computer simulations? Denning (1990) reviews discussions at an Association for Computing Machinery (ACM) meeting, November 1990, that provide some of the key arguments for and against this question. Arguments that caution against models included the following:

> In most socio-economic domains, models have not proven as trustworthy as human experts (Dreyfus 1990). The approach used by experts is likely an uninterpretable brain activity evolved through formal encounters with teachers.
>
> Systems that involve humans must capture human responses. This makes the rules of the system dynamic because of the self-reflecting nature of people.

Arguments in favor of modeling reported by Denning include:

> Individuals are "notoriously inept at understanding the dynamics of systems that contain feedback". Most mechanical, organizational, social, and biological systems contain feedback loops (Forrester 1990).
>
> The systems are too complex for human experts to master. Simple physical systems like fluid flow contain on the order of ten equations. Hardware systems like airplanes or computer networks are described with 106 equations (Kline 1990).

Currently, formal expressions of landscape models are being captured in computer models by scientists and land managers across the world that believe in the promise of modeling.

Today, the process outlined in Figure 1 is beginning to change with the design and development of simulation models that represent some aspect of the landscape dynamics. Figure 2 suggests that some of the conceptual models are being translated into formal computer simulation software. This is being accomplished by developing simulation models that explore general principals of nature. In Figure 2, two of the conceptual models of Figure 1 have been captured in software. Now (1) teams of scientist can jointly develop complex models based on their collective knowledge of the system (2) the model can be evaluated, and (3) it can be extended and attached to other models. This emerging approach to landscape management still requires a political process to integrate the output of the individual models with each other and with the remaining "output" from the conceptual models. I-STEMS seeks to complete this transition by providing a software environment that allows the development of multi-disciplinary simulation models.

Figure 3 suggests that the step missing from the full transformation from conceptual modeling to computer based simulation modeling is a collaborative modeling environment within which all professionals associated with landscape management can create integrated spatio-temporal ecological models. The separate hydrology, plant succession, land activities, habitat suitability, and other models can be designed and developed in a manner that allows each to run landscape simulations in synchrony with each other. This document describes a system that meets the requirements necessary to realize such a capability.

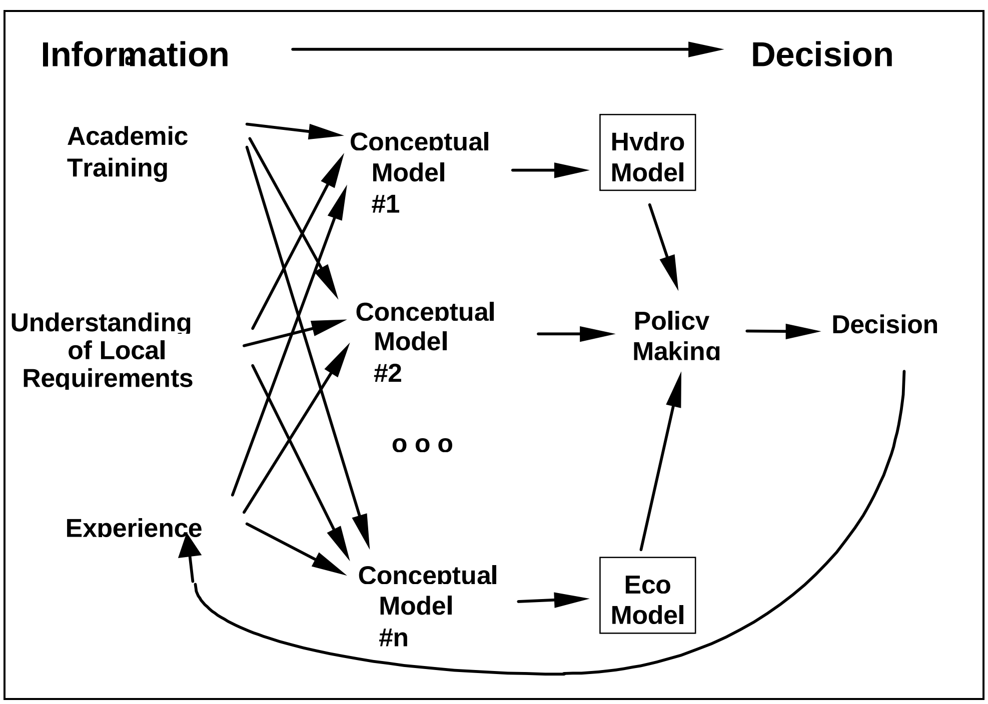


Figure 2. Emerging approach to landscape modeling.

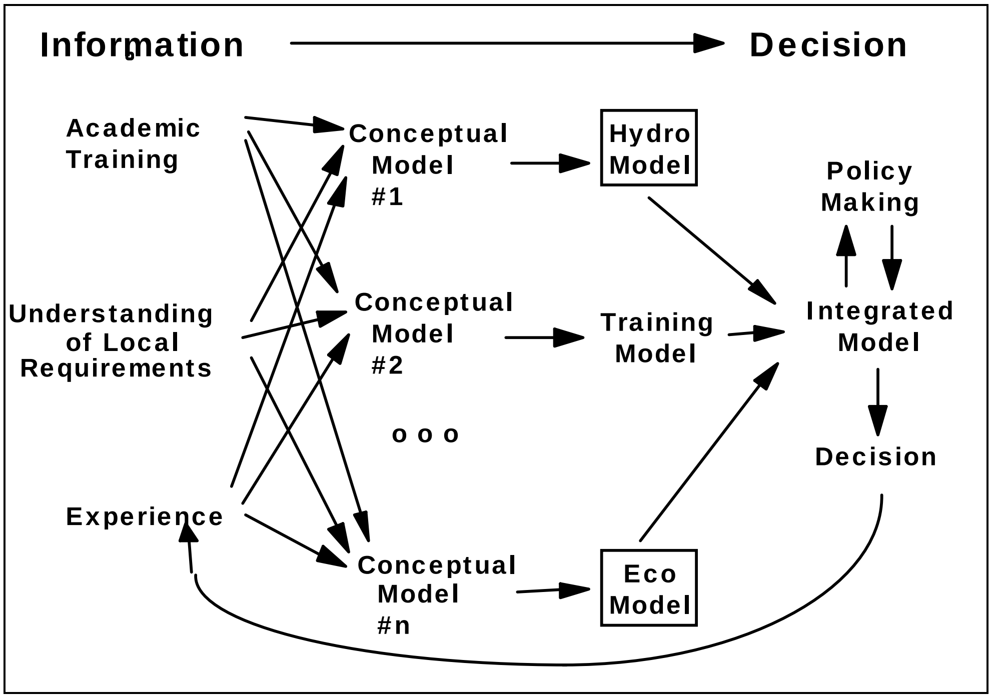


Figure 3. Future approach to landscape modeling.

4 Design Philosophies

Design and development of a software environment is a complex undertaking from many different standpoints. For this effort a large number of design goals must be recognized, considered, and addressed. They include goals that must be addressed by a collaborative interdisciplinary team which includes target end users, ecologists, economists, statisticians, simulation specialists, mathematicians, and computer programmers. Note that computer programmers are put at the end of this list not to minimize their importance, but to highlight the importance of the other players who sometimes can be neglected in software development projects. The key design goals/philosophies are presented separately in the following paragraphs.

Embrace Current Ecological, Economic, and Management Theory

I-STEMS is about integration; it is not an exercise in reinvention. This is especially true for simulation software in support of ecological, economic, and management components of the system. I-STEMS is envisioned as a state-of-the-art land management tool. As such it must recognize and embrace current theories that are being used today. Although new theories, concepts, and ideas are emerging, the land manager typically is looking to rely on systems based on the best available theories. Although I-STEMS can be used to test and develop new theory, its focus will be on the application of concepts and ideas that have proven to have the best predictive capabilities.

Embrace Historic Software

Existing, proven, and well-appreciated landscape simulation software must be selected for inclusion into an anticipated growing family of I-STEMS software. Embracing legacy (already developed) software allows the I-STEMS Research and Development (R&D) team to focus on a more narrow set of design and development issues. Redeveloping working capabilities does have the advantage of allowing the software programmers to better integrate disparate software, to optimize the algorithms for exploiting a particular set of hardware, and to create better consistency between various pieces of software. These benefits come not only at the cost of redevelopment, but also at the loss of potential collaborations with the developers of legacy software. Finally, legacy software typically has developed a following of individuals who form a pool of new customers who have been satisfied with the results from that software. As such, I-STEMS will initially embrace legacy software. Examples are the GRASS GIS, the Spatial Modeling Environment (SME), the agent-based simulation environment SWARM, and the CASC2D overland storm-flow simulation software.

Modular

Today software programming embraces modularity for design and development. Object-oriented programming defines today's preferred approach to modular programming. For I-STEMS, modularity is a goal for all levels of design. The system will rely on only a small set of components to function. Most components will be optional, replaceable, and interchangeable with other components. How the components are mixed will depend on the particular simulation being developed for a specific end user.

To be modular, strong standards will be developed to ensure the mixing of different components developed by different I-STEMS teams. If a particular graphical user interface (GUI) is to be effective as a viewer or controller for any number of subsystems, those subsystems and the GUI must be developed to a common set of specifications. A cornerstone of those specifications will be the requirement for system components to be individually functioning objects that run as separate programs on a network, yet are connected to other objects through interchanges of information across the network via standardized protocols.

Distributed

Dynamic, spatial, ecological simulation models can rapidly become very complex and can overwhelm single-processor computer systems. It is important that the simulation models eventually assembled for addressing real landscape management questions and concerns be able to use any number of available processors. These processors may exist within a single machine or may be distributed across a heterogeneous network. Distributed processing will be accomplished through a variety of means. First, as noted above, system components will effectively be independently operating programs communicating with one another in a heterogeneous network of computers. Because the components operate as single programs, they can easily be distributed across any number of processors and computers. Second, some modules will be developed that also make use of specific parallel processing environments.

Multiple Interface Levels

I-STEMS will be developed with at least three user interface levels in mind (see Table 1). The I-STEMS software developer will work with well-defined application programmer interfaces (API). These will include all of the standardized system objects, routines, and data exchange methods to support (1) the encapsulation of legacy software simulation components and (2) the efficient design and development of new components. Model developers will then use the I-STEMS modular model components to create location- and management-specific models to be used as landscape decision support systems. The models they create will be used by land managers for risk assessment, analysis of impacts, and improvement of land management techniques and schedules.

Model Components as Objects

I-STEMS will embrace object-oriented software design approaches. Design and development of objects is more expensive than traditional programming approaches. Also, execution time for software developed with objects can be significantly slower. The payback occurs with the ability to rapidly recombine sets of objects. Because each object is essentially a self-contained program, it can be combined with other objects without conflicts with other software. Each appears to the other objects as a “black-box” with potential inputs and potential outputs. The internal operations of the object are hidden from other objects.

Table 1. Three levels of system interface.

Interface Level	Activity	System View
Software developer	Encapsulation of legacy software into subsystem objects. Development of new simulation software modules.	Application programmer interfaces Software modules that run as stand-alone programs
Model developer	Develop models for end-user resource managers	Libraries of system components Configuration files Assorted viewers and controllers
Resource manager	Manage landscapes with respect to mission goals	Any number of simulation models A consistent interface between models

To the future model builder I-STEMS will offer a family of independently developed simulation objects. These will include any number of landscape simulation modules that may be object encapsulations of simulation models before I-STEMS. For example, GIS operations, hydrologic simulations, plant succession models, and weather simulations will be captured in such modules. I-STEMS will be an open software environment within which any research group will be free to design and develop additional simulation modules.

Encapsulation of models and simulations will be accomplished in a very rigorously defined manner that will ensure that every module will be as broadly recognizable to other objects as possible. The consistency in external appearance of simulation modules will then allow the design and development of user-oriented visualization and control objects.

Target Hardware/Software

It is anticipated that the research, design, and development efforts associated with the initial prototyping of I-STEMS will occur between 1996 and 2000. As such, the following hardware and software will be adopted:

- *UNIX*: This operating system currently provides the best opportunities for software research and development. Software and ideas are shared freely in this historically research-oriented environment. The version of UNIX selected is Sun's Solaris 2.5. This environment supports multiple processors at the operating system level, which provides a good level of parallel processing capabilities.

- *C++*: This language is used by many of the software programs that may contribute to the I-STEMS R&D efforts.

- *CORBA*: The Common Object Request Broker Architecture specification will be used to provide the fundamental communication channels between disparate landscape simulation software modules running as separate processes.

During R&D, these restrictions will be continually reviewed and refined. Software and hardware requirements for final products will be established with respect to needs and available solutions.

5 Land Manager View

The sole purpose of I-STEMS will be to support landscape managers confronted with a variety of short- and long-term goals and initiatives that involve information associated with a large number of academic disciplines in the natural sciences as well as political, economic, and legal requirements. Dynamic landscape simulation will commonly help address the following types of land management decisions by allowing the managers to look more clearly into the future than what is currently possible.

- *Past, present, and future landscape schedule.* Displays will show concurrent and potentially conflicting activities. Input opportunities will allow users to seek time slots and space opportunities that meet a set of criteria.

- *Immediate environmental impacts anticipated.* By using traditional GIS overlay analysis capabilities, users will be able to see immediate impacts resulting from scheduled events.

- *Predicting cumulative and indirect impacts.* Users will be able to predict cumulative and indirect effects associated with scheduled activities. Optionally, the cost of each scheduled event can be computed.

- *Planning long-term land-use patterns.* Users will be able to analyze alternative patterns with respect to intensity of activities that can be sustained, biodiversity, ecosystem health, economic costs, and impacts on Threatened and Endangered Species (TES).

Audience

The view of I-STEMS presented here is that which will be seen by typical land managers. These people typically are required to respond to a continual stream of requests which, though seemingly unpredictable on a daily basis, fall into a number of distinct categories over the long term. Land managers work at the intersection between the variability of nature and the scheduled predictability of human activities. Their job requires the ability to respond to nature in a manner that minimizes the impact on the human expectations of schedules and plans. They must predict changes in the natural world due to the activities of humans in order to effectively manage the landscapes and species for which they are responsible.

System Design Philosophy

The typical land manager will not actually see I-STEMS. Although the acronym will be familiar, it will be an array of land management decision support systems (DSS) that provides a familiar face to I-STEMS. These systems will be known as I-STEMS models and, for any given location, may be quite numerous. Some models will attempt to be complete simulations that capture all aspects of the landscape processes simultaneously. Others will focus on very specific management questions. A large model might simultaneously simulate landscape activities running at a number of different spatio-temporal scales to simultaneously capture such components as the behavior of individuals representing a threatened or endangered species, the behavior of larger populations, human activities including training, logging, recreation, economic consequences, biodiversity consequences, fire, disease propagation, and movement of genetics.

Models will have a similar look and feel for they will be constructed from a common toolbox of software. Models will be controlled and viewed through a set of standardized, human-computer interface components. Once a manager has become comfortable with a particular model or two, other models will automatically feel familiar.

Human plans occur at a number of different scales in time and space, and they interact with nature at each scale. At one end of the land-management spectrum is the emergency response, which involves responding to system breakdowns and shifts in nature that occur at relatively short time scales (hours to weeks). At the other end of the scale, the land manager must consider how human activities and schedules interact with long-term natural processes such as hydrologic systems, soil maintenance, population succession, dynamics, and even evolution. I-STEMS will provide tools and functionalities that allow for the construction of models at both extremes individually or together in a hierarchical context.

Multiple Models

It is anticipated that land managers will eventually use a number of different models to help them manage landscapes. The ideal situation, of course, would be for a land manager to have a single interface within which any land management scenario could be evaluated with respect to all important consequences. Suppose one does some “blue sky” musing and describes this ideal system. First one must identify the types of land management decisions that the system will need to evaluate. For example:

- Layout of buildings, roads, and training areas
- Schedule of land use
- Schedules for land rehabilitation.

Next one must identify the types of questions that a land manager will want to pose to their system. The following list may cover the range of questions.

- Project the land-cover anticipated during a season ... or during a decade
- Project and evaluate the biodiversity anticipated over a century
- Anticipate the cost of each scheduled training exercise with respect to environmental damage
- Estimate the anticipated impact on TES
- Assess the burn potential during the year
- Anticipate the erosion potential during scheduled training exercises
- Compare different training schedules with respect to multiple objectives
- Optimally schedule a list of training activities within scheduling constraints
- Analyze the landscape’s adaptation, resistance, and resilience to disturbance, including fire, disease, storms, and flooding.

These issues require analysis at a number of different spatio-temporal scales. It is not reasonable, with anticipated hardware and software over the next 10 years, to address all of these questions and objectives within a single simulation model. A small number of models might be developed to cover the range. Each model would focus on processes that occur at similar time and space scales. For example, I-STEMS might be used to develop the following models.

Emergency Simulation and Analysis (ESA)

This potential system could focus on rapidly changing dynamics associated with emergency situations. It might have the following subsystems:

- Wildfire simulation
- Chemical spill
- Storm and flood simulation.

Scales:

- Time step: minutes
- Time extent: days
- Spatial resolution: 1 to 10 meters
- Spatial extent: subtraining range to local.

These models would be run by environmental office personnel to help guide emergency responses to unusual situations. ESA would initialize a simulation by extracting the current state of the landscape from another simulation. It would also be used to simulate the potential for an emergency situation.

Installation Seasonal Simulation and Information System (ISSIS)

ISSIS could provide a simulation environment that would simultaneously be used, on a daily basis, to (1) keep track of the current state of the landscape, and (2) project the state of the landscape over the current season.

Scales:

- Time step: 15 minutes to 1 day
- Time extent: 1 to several years
- Spatial resolution: 10 to 100 meters
- Spatial extent: local.

Inputs:

- Range control
  - Training schedules
  - Tables of Organization and Equipment (TOE)
- Environmental office
  - Land rehabilitation
  - Impact model input
  - Measurements of landscape health.

Outputs:

- Land cover predictions
- Environmental cost of each training exercise
- Anticipated erosion potential
- Comparison of different potential schedules
- Anticipated impacts on TES and critical habitats
- Changes to habitat suitability indices for selected species.

This system would be developed for daily use by personnel in the environmental and range control offices. Each office would be responsible for managing certain inputs. Any office could then use ISSIS to project the landscape into the immediate (1-year) future. ISSIS would need to interface with other management systems in daily use like the local GIS, the Range Facility Management Support System (RFMSS), and others.

Installation Regional Effects Simulation System (IRESS)

This hypothetical model focuses on the long-term (years to centuries) consequences of land management patterns. It would be used primarily by environmental offices to address long-term consequences of land-use patterns, forestry, and regional landscape patterns with respect to biodiversity, sensitive habitats, TES, and successional states of the land.

Scales:

- Time step: 1 month to 1 year
- Time extent: decade to century
- Spatial resolution: 100 to 1000 meters
- Spatial extent: local to regional.

Inputs:

- Range control
  - Range configuration
  - Anticipated use patterns (in time and space).
- Environmental office
  - Forest management plans
  - Ecosystem response models
  - Successional models

– Land condition trend data.

Outputs:

- Landscape successional state projections
- Long-term TES and habitat suitability index (HIS) potentials
- Biodiversity predictions (regionally oriented)
- Comparison alternative schedules
- Anticipated impacts on TES and critical habitats.

Each of these hypothetical systems would be constructed within I-STEMS, but the users of the system will not actually be working with I-STEMS. Because the systems are developed within the same environment, they will provide consistent interfaces to the end user.

When installation standard models (e.g., ISSIS, ESA, and IRESS) are not sufficient, an installation can turn directly to I-STEMS to design and customize a new simulation model.

Model Modification

Simulation models run by installation personnel will be associated with a large number of input options. These can be divided into initialization and run-time parameters.

Initialization

Landscape simulation models must be initialized with the starting state of the system. Initialization will likely involve:

- Landscape maps that identify such things as vegetation type and density, topological information, and land ownership
- Schedules of landscape activities

- Weather statistics
- Tables of Organization and Equipment
- Training activity descriptions.

Run-Time Parameters

When models are run, a number of options can be provided by the modeler. These include:

- Assignment of subprocesses to computers
- Identification of how the model will be visualized during and after a simulation
- Identification of run-time input options
- Debugging output options.

6 Modeler View

Audience

This chapter describes what the individuals developing new simulation models will see when working with I-STEMS. The development of landscape simulation models requires the coordination of an interdisciplinary group of individuals. It is presumed that these individuals will not, for the most part, have the skills necessary to design and develop new simulation modules using low-level software languages. They will, however, be assembling simulation modules into complete landscape simulation models. The next section, “Imagine”, describes the development of a simulation model by an interdisciplinary team. This is followed by discussions of the system design philosophy as viewed by a modeler (p 43), the model control center (p 43), subsystem examples (p 46), and generic viewers and controllers (p 48).

Imagine

To help visualize the utility of the I-STEMS geographic modeling system, imagine a future scenario that involves a simulation challenge at a military installation. The year is 2003. Fort Hood, TX has been challenged to expand its training areas into adjacent properties. This expansion is desired to accommodate an expanded tracked vehicle training mission. The environmental office is tasked with generating several annual training scenarios and then evaluating each with respect to the direct and indirect impacts on: (1) the ability to train (2) Golden-cheeked Warbler populations (3) Black-capped Vireo populations, and (4) local and regional biodiversity. This effort is part of the environmental assessment (EA) requirements. Management decides that the analysis shall be accomplished by an interdisciplinary group consisting of individuals from the environmental, training, and scheduling offices. They will have at their disposal several workstations that have recently been used to test the latest version of I-STEMS, the Integrated Spatio-Temporal Ecological Modeling System.

Days 1and 2 — Team Assembles

A high-priority meeting is held to assemble and brief the team that will be handling this assignment. They are tasked to develop a dynamic training area simulation model focused on the intended expansion area and adjacent existing Fort Hood properties. The resulting model will be used to evaluate the direct and indirect impacts of a change in mission on these areas. Several concerns need to be addressed before this area can be used for the intended training: (1) two threatened or endangered species (2) potentially sensitive ecosystems and habitats (3) water quality requirements for drinking water wells down-stream, and (4) impact on regional biodiversity initiatives, and (5) cattle grazing goals. The team must provide a working simulation model and preliminary results within 20 days to support a briefing to visiting dignitaries. In addition to the workstations at each member's desk, the main server located in the environmental office is available (a $50K machine containing 256 Mbytes of internal RAM, with four 200-MHz processors, and 20 gigabytes of on-line hard-disk). Fort Hood has been connected to the Internet since the mid 1990s and now has a 10-megabit-per-second connection to the outside world, which provides them with powerful run-time access to several supercomputer centers including the thriving National Center for Supercomputing Applications (NCSA). The team will be using the latest release of I-STEMS, the Integrated Spatio-Temporal Ecological Modeling System.

Following the initial briefing, the team meets and establishes the following subteams:

- Species-specific models
- Weather and climate
- Hydrology
- Communities and ecosystems
- GIS and image processing
- Visualization and control
- Training.

Each team is tasked with identifying and evaluating local sources and available model components distributed across the network.

Day 3—Available Component Reports

The simulation team meets to brief each other on the information discovered during a day of exploration. Potential system components are presented in Table 2. All components conform to the I-STEMS standards, which allow them to be readily integrated. Team reports are:

- *Species-specific Team*: Three models of local threatened and/or endangered species are available. Population- and individual-based models are available for the Black-capped Vireo and the Golden-cheeked Warbler. The team recommends adopting the population-based model.

- *Weather and Climate*: Weather and climate models have both been identified on the network. Both are identified as standard, accepted models and model outputs.

- *Hydrology*: The Saghafian (Saghafian 1993) model was located in I-STEMS format. It has now been verified on a wide variety of landscapes. Also, a new soil compaction model conforming to I-STEMS has been located on a server at the U.S. Army Engineer Waterways Experiment Station (USAWES).

- *Communities and Ecosystems*: The standard Army-developed plant succession model is available in Beta release 4.2 form.

- *GIS and Image Processing*: Extensive historical and current geographical information system and imagery data exists on-site and can be adapted to I-STEMS.

- *Visualization and Control*: The Internet server at the University of Illinois currently offers a wide variety of visualization and control objects for I-STEMS applications. These include the traditional meters, sliders, menus, feedback panels, dials, and buttons. This site also makes available several sophisticated new intelligent controllers that manage tradeoff options, various optimization approaches, and collaborative modeling tools.

Table 2. Hypothetically available model components.

Potential Component	Source	Description	dT,dS	Inputs Required	Outputs Available
Black-capped Vireo	USACERL	Population model object developed for Ft. Hood (1998)	1 wk, 1 km	Weather Topology Vegetation (grass, forb, shrub, tree)	Densities in 6 age classes
Black-capped Vireo	U of Texas	Individual based object developed for the State of Texas (2001)	1 dy, 100 m	Density of predators Weather Topology Vegetation (5 species)	Location Health indices (5) Age, sex, etc.
Golden-cheeked Warbler	Texas A&M	Population model object developed for Ft. Hood in (1998)	1 wk, 1 km	Weather Topology Vegetation (grass, forb, shrub, tree)	Densities in 6 age classes
Vegetation density maps	USACERL	Vegetation, grass, shrub, forb, and tree (2002)	N/A, 30 m	N/A	N/A
Vegetation succession model	Colorado State Univ.	20-species succession model (1999)	1 mo, 100 m	Soil type Soil compaction State of starting vegetation	Succession phase
Tracked-vehicle impact model	USAWES	Soil compression model (1997)	N/A, N/A	Tracked-vehicle days per ha Soil type	Soil compression
Biodiversity model	INHS	10-keystone species model (1998)	1 yr, 10 km	Climate % land in each of 5 succession states	Densities and genetic variability for each species
Training models	USACERL/ Ft. Hood	Maps created for each exercise and training area combination (2003)	1 day, 30 m	Training exercise Training area	Average tracked-vehicle days per HA.
GIS	Ft. Hood	Extensive 100+ theme digital map data base	N/A, 5-100m	N/A	100+ themes, some historical data; extensive imagery.
Hydrology	USACERL	The Saghafian finite-difference model (Saghafian 1993)	minutes -days, 30 m	Topographic data, land use and cover	Saturation, depth, velocity, scouring and deposition
Weather	National Weather Service	Historical and average weather conditions and probabilities	1 day, 100 m	Day of year	Temperature and rainfall: average, standard dev, and probability

- *Training*: Two training model sets are available. Fort Hood's training impact tables have been used quite successfully for the past decade and relate training exercises and training areas with degrees of estimated environmental damage. USACERL's relatively new set of maps add a spatial dimension to these tables and provide impact information at a resolution of 30 meters. The team decides to adopt these maps and the USACERL approach to developing such maps.

Day 4—Register Available Submodels

A new model is established on the server. This process consists of setting up an information exchange server that will facilitate communications between different processes running on different machines. All participants are told to establish I-STEMS environments on their individual workstations, which attach to this server. Once this is done, all team members can readily query and view any portion of the developing model as well as establish model components on their own machines. Team members then begin to set up the submodels selected from Table 2 on the local machines. By the end of the day, each member is able to view the status of the virtual interconnections between the various submodels. For example, a query on the status of the hydrologic simulation model yields the report shown in Figure 4.

This report begins by indicating that this submodel has been registered with a "main model" called "Ft. Hood Extension Simulation," which is registered on the machine called env.fthood.army.mil. Connection to this model is accomplished with the code: 175 (which is a port or socket type number). This submodel has registered itself with the main model and will be running on and accessible through hydro.fthood.army.mil. Note that all submodels may run on separate machines. Underlying information brokers facilitate virtually seamless integration of these submodels. Some model metadata is also displayed. Here, that information identifies the version number of the submodel and the latest I-STEMS version under which the model is known to operate. A section on inputs and outputs provides information on how the submodel is currently linked to other submodels. These links were established using user interfaces that probe the model space for available variables and then allow the modelers to establish the desired connections. The CONVERTER column under inputs identifies which, if any, standard unit converters were used to establish the connection. The Outputs section identifies the submodels that currently use the available outputs. The lists of such submodels will grow and shrink as the different components link themselves with each other.

The input “water” is identified as being supplied by submodel “Dummy.” This is a reserved submodel name that is attached to very simple data generators. Model developers are allowed to create dummy inputs defined by fixed values or graphs that use time (e.g., month) as the independent variable. The purpose is twofold. First, inputs that are not being generated by other submodels can be simply accommodated in this fashion. Second, during debugging and sensitivity analyses, input variables can be set to static values.

MAIN MODEL INFORMATION

Name:	Ft. Hood Extension Simulation
Main Server:	env.fthood.army.mil
Access Code:	175

SUBMODEL INFORMATION

Name:	Hydrologic Simulation
Model Server:	hydro.fthood.army.mil
Access Code:	180

METADATA

Author:	Bahram Saghafian
Version:	4.3.1
I-STEMS version:	2.6
Resolution:	30 meters

INPUTS

NAME	UNITS CONVERTER	INITIATED BY	SUPPLIED BY
Elevation	meters	GIS	N/A
Slope	percent	GIS	N/A
Initial saturation	mm	GIS	N/A
Soil permeability	mm/day	GIS	N/A
Manning’s K	K	GIS	Vegetation Model
Water	mm/hr mm/inch	N/A	Dummy

OUTPUTS

NAME	UNITS	USED BY SUB MODEL
Soil saturation	mm	Vegetation
Water depth	mm	Vegetation Golden-cheeked Warbler Black-capped Vireo Training
Water velocity		Vegetation
Soil scour/deposition	mm	Vegetation, Succession

Figure 4. Hydrologic simulation model report.

Each simulation model can be asked to generate a simulation report. The most important classes are viewers and controllers. Viewers are essentially submodels that only access output from other models; they probe submodels and display information in numerous fashions. Generally this means that they provide run-time views of system states (maps, tables, strip charts, etc.) or dump data to output files for later analysis. Controllers, similarly, are basically submodels that provide input from people to submodels. Based on human interactions with graphical user interfaces (GUIs), they supply values to submodels. Such inputs are injected into the associated submodels at the time they are set (typically the receiving submodel controls the data probe).

Of the numerous other interfaces available to the modelers, two require brief mention here. A main control panel is available for starting and controlling the model, as a whole. This interface allows the user to turn any of the various submodels into ON, OFF, and STATIC modes. OFF makes the submodel appear to be nonexistent. STATIC turns the submodel off, but allows it to generate predefined static information much like the “Dummy” submodel. ON causes the submodel to operate normally during the course of a simulation run. These are used to control the view and controller components as well. The second general type of important interface is the control panel for supporting simple modifications to each of the submodels and view/controller interfaces. For example, a generic population submodel can cover a wide range of populations by simply allowing the modeler to “tweak” such attributes as growth rate, consumption rate, fecundity rate, or home-range size. Alternatively, a user interface might allow a wide variety of displays for a given series of data: bar chart, strip-chart, colors, or ranges.

Days 5-10—Research To Develop Missing Components and To Extend or Modify Available Components

The team uses a full week to followup the initial assembly of available components with some development of additional simulation components. In particular, the available visualization tools have to be assembled in a manner that maximizes the match to the current application. For example, the training submodels need to be upgraded to reflect the new training scenarios and weapon systems anticipated for the new landscape. Each submodel is run independently to identify as many potential errors as possible.

It is also decided that two models will be developed to help address the overall goals and objectives. The biodiversity questions require a time-step and resolution sufficiently different from the other questions to warrant a separate model.

Days 11-14—Integrate and Debug

This week's effort involves numerous runs of the simulation model with successively more components turned on. As conditions are discovered to move out of reasonable ranges (negative populations, temperatures over 150 °F, and succession stages out of line with simulated training), errors in the submodels and data are discovered and repaired. Sensitivity analyses are conducted on the more uncertain inputs — some of which are found to be quite important. The developed user interfaces are also tested and improved to remain stable.

Days 15-20—Management Evaluation of Alternatives–Reports Generated

During the final phase, management representatives are invited to participate in the final simulation runs. Some different training schedules are run along with some updates to potential property boundaries and road network possibilities. Output videos are generated for playback at future meetings and are captured for viewing on the Internet. It appears that more of the objectives than first imagined can be met through newly recognized arrangements of the planned training activities. Key locations, thresholds, and leading indicators are identified for particular monitoring as a strategy is implemented. The models are documented and made available to the management team for use in making decisions within the chosen strategy.

This imaginary scenario, based in some fact, suggests that a geographic modeling system will be useful to landscape managers (here military installation training range managers) for the rapid design and development of location specific dynamic simulation models. These models will simulate various components of the landscape, simultaneously using appropriate spatio-temporal scales for each. Long-term and indirect effects and interactions between the various components will be available for managers and other decisionmakers to explore.

System Design Philosophy

The previous story suggests a number of design philosophies that are explicitly stated here. First, the modeling environment supports difficult *collaborative* efforts. Specialists are each assigned to peruse libraries of I-STEMS compliant models to analyze and identify potentially suitable submodels. These submodels will have been designed and developed by specialists (e.g., hydrologists) for the purpose of being later connected with submodels developed by other specialists (e.g., range or plant succession scientists). Submodels reflect the philosophy of *modularity*. Each submodel used in the story was developed outside of the final model being assembled. Each is a standalone object that is prepared to behave and interact with other submodels developed at any number of research and development sites.

Second, from a computer science standpoint, the assembled model runs in a *distributed* and perhaps heterogeneous computing environment. Individual submodels will be allowed to run on platforms for which they were developed while simultaneously interacting with other submodels running on different CPUs and even different machines within a local or wide area network. Each submodel will be developed to communicate with other submodels using *standardized intercommunication protocols*. One class of submodels will be *viewers* and *controllers*.

Model Control Center

An I-STEMS model consists of a number of key components, each potentially running on a different CPU or a different machine on the network. From the modeler viewpoint, these components can be grouped into the following two categories:

1. The model control center

2. The model subsystems.

The control center is discussed here while the model subsystems are described, from the modelers viewpoint, in the next section. A control center consists of a number of interrelated programs that together provide the environment for initializing and managing the subsystems. It is associated with a user interface that provides various viewports into the operation of the full model. The control center has two primary responsibilities. First, it maintains information about the various submodels being used. This includes model name, machine to which it is assigned, data it requires for initialization and input, and data it can provide during a simulation. For example, Tables 3 and 4 display sample input/output information for two hypothetical submodels. If these two submodels were instantiated by the control center, Table 5 Submodel Data Exchange, could be displayed automatically to identify to the modelers the current match between required inputs and available outputs. Associated with each data stream will be data units, associated error information, and frequency of data changes. Second, the control center will monitor system and model performances during a simulation. This may include CPU usage statistics on the various machines and rate of data exchange between submodels (especially between machines across a network).

Table 3. Hydrology submodel (sample).

Submodel Name:	**Hydrology**	
Variables available for output		
	Name	Units
var	Water Depth	cm
var	Soil Saturation	%
Variables required for input		
type	Name	Units
fixed	Soil Permeability	
fixed	Soil Depth	meters
var	Manning's K	K
var	Rainfall	mm

Table 4. Vegetation submodel (sample).

Submodel Name: Vegetation		
Variables available for output		
	Name	Units
var	Percent Live Veg Cover	%
var	Percent Dead Veg Cover	%
Variables required for input		
	Name	Units
var	Soil Saturation	%
var	High daily temperature	°C
var	Low daily temperature	°C

Table 5. Submodel data exchange.

Model Variable	**Initialized by**	**Managed by:**	**Used by:**
Water Depth	?	Hydrology	
Soil Saturation	?	Hydrology	Vegetation
Soil Permeability	?		Hydrology
Soil Depth	?		Hydrology
Manning's K	?		Hydrology
Rainfall		?	Hydrology
Percent Live Veg Cover	?	Vegetation	
Percent Dead Veg Cover	?	Vegetation	
High daily temperature			Vegetation
Low daily temperature			Vegetation

Control centers will allow modelers to assemble model components while monitoring the interactions possible between submodels. This will be accomplished in networked environments by "slaving" remote control centers to a master control on a selected computer. Each control center will have access to local tables of available submodels and will be able to instantiate these models as directed by the user operating the master control center. As different submodels are "brought-up," they announce their data requirements and offerings, which the master control center manages and optionally provides to the operator. Once a set of submodels is initialized and have all of their input data requirements accommodated, the control center can set, and then start and stop, the master simulation clock. During simulation runs, optional viewports may display run-time statistics.

Finally, control centers will have the option of saving the parameters associated with a simulation (complete or incomplete) in master files. These files can be used to fully initialize a simulation model at a later time with minimal user interaction.

Subsystems

As noted above, a fully operational I-STEMS will provide a toolbox of submodels designed and developed at numerous research and development centers. These will be accessible through libraries constructed on the Internet. Each submodel will be associated with metadata describing the characteristics of the model and containing refereed reviews of the model identifying conditions for which the model is and is not useful. Next to a growing library of submodels, the I-STEMS "core" will be relatively small — providing only standards for submodel design and development that will ensure interactions with other models through well-defined communication channels.

Common "Appearance"

Each I-STEMS compliant submodel will interact with other submodels via standardized protocols captured within an application programmer interface. To the model developer working directly with compliant submodels, this will mean that each submodel will offer a common and consistent appearance to other model components. For example, when a submodel is initialized at run-time, it registers, with the associated control center, information it requires to operate as well as the information it can provide. Information may then be readily displayed by the control center covering all participating submodels (e.g., Table 3: Hydrology submodel (sample)).

One set of model components will be a standard set of viewers and controllers (discussed later in some detail). Because of: (1) their repeated use between models, and (2) their being the only interface between people and the submodels, the viewers and controllers will provide the most visible consequence of consistence in "appearance" of submodels to each other.

Submodels Are Software Plus Data

Submodels will be designed and developed as independent objects. An object here is defined as a standalone set of data and software instructions. Object-oriented software approaches have been adopted by many software developers. This has resulted in the development of a wide range of software programming languages such as C++. I-STEMS brings this development paradigm to the model assembly level.

Those unfamiliar with the paradigm may benefit from a short explanation of the meaning and consequences of object-oriented programming. Many I-STEMS model developers will be familiar with the use of geographical information systems. These systems have traditionally distinguished between the data associated with a particular landscape and the software (the GIS) used to create, display, and manipulate that data. To query or analyze a map (or maps) the GIS operator would invoke the local GIS to perform the query or analysis. An object-oriented GIS would combine the operations with the maps. This combination would “exist” on its own — separated from other processes. A person could then ask the object to perform the desired query or operation on itself.

For example, using traditional GIS reasoning, a user would ask a GIS package to invoke a particular program with certain user inputs on a map (or set of maps). For example one might start-up the GRASS GIS and run a command like:

```
r.info soils
```

This is a request to run the r.info program on the map called “soils.” In an object-oriented GIS, the syntax is reversed. Instead of asking the r.info program to process the soils map, the soils map is asked to provide information. The command might be:

```
ASK soils TO GiveInfo
```

The key difference is that what had been an inert piece of data is now an active entity, capable of responding to certain requests. This changes the way data is viewed and opens the opportunity to integrate digital landscape information in more dynamic (rapidly changing) ways. In a traditional GIS, a process is invoked on a map. This requires that, at a minimum, the map be read into memory, the data be processed, and the map be written back out to disk. Each procedure performed on the map, regardless of the complexity, goes through these processes. If there are to be many operations on a map interspersed with operations on other maps or files, the concept of a map as a “living” object becomes attractive. A map object can keep a map active for the duration of a simulation or series of different operations. A simple map object may be able to pull a map into virtual memory and then respond to a request for changes to or information about the map over time.

This object-oriented concept is especially useful for dynamic landscape simulations. Essentially, each I-STEMS submodel is associated with the state and management of certain landscape information — certain maps. Conceptually then, I-STEMS is a dynamic simulation-focused, object-oriented GIS. The model developer will think about landscape simulation as the assembly of interacting dynamic maps. For example, the vegetation-cover-map, is actually a dynamic simulation of the vegetation, which might use rules based on Clementsian plant succession. A training-simulation-map might represent a training exercise complete with its mission, materiel complement, and limitations on fuel, time, and allotted environmental impact. The I-STEMS model developer must think of submodels as objects that combine behavior rules with system state information. The processes and the data are combined into objects that will grow into extensive libraries.

Viewers and Controllers

One class of I-STEMS objects will become very familiar to any I-STEMS modeler and, effectively, any manager running complete I-STEMS models: the viewer and controller object class. As discussed above, an object in object-oriented programs consists of information (data) and operations. Objects respond to requests from external objects and may invoke requests on external objects. In the case of viewers and controllers, the “operations” are provided by human operators. That is, conceptually, the human operator is viewed by other system objects as existing inside the viewer and controller object. An object has no knowledge of what goes on inside any other object; it only knows that it can make certain requests of the object. That a human or a computer automata resides in the object is of no consequence to other objects.

In I-STEMS, viewer and controller objects provide the only user interface. The I-STEMS rule will be that software within model objects do not drive peripherals (monitors and keyboards). The reasons for this are threefold. First, consistent user interfaces can be better maintained and managed if each submodel is not allowed to perform its own interface with operators. Second, of all software written, the user interface has the shortest relative life span. I-STEMS submodels are expected to enjoy 10- to 30-year life spans while the interface software is expected to be viable for only 5 to 10 years. By forcing the separation of the user interface from the models, I-STEMS will be more efficiently upgraded over time. Finally, software must not be written to require specific peripherals. Submodel developers forced to use established viewers and controllers are less likely to write system-specific software. Four classes of viewers and controllers are suggested below. There are, however, potential combinations of these and others as well.

Run-time Visualization

A pure visualization object submodel simply probes, during a simulation run, certain user-selected information available from the submodel objects. The methods (software calls) that visualization submodels use are identical to the calls model submodels use to query one another. Run-time visualization objects will be developed to provide a number of viewports into the operating model. This will include:

- Map views that might overlay raster, vector, and point information
- Time-series views of selected state variables in “stripchart” formats
- Tabular views of selected state variables
- Views of overall system status including the load on computational circuits (CPU, network, memory, and disk).

Run-time Control

Control submodels provide operator run-time input options to a simulation. Simulation control will include:

- *Control of the overall simulation*. This might include the ability to start and stop portions of the simulation or the ability to exchange one submodel for another.
- *Control of individual submodels* such as adding or deleting components, or changing the state of the submodel.

Perhaps most control submodels will also be viewer submodels; viewing and controlling are conceptually two sides of the communication process. It will not be unusual for a viewer to be used without an attached controller.

Data Storage

As a simulation runs, the state of the simulation is continually changing. A complex simulation typically cannot retain the complete state of its system throughout the entire simulation run. Consider a landscape represented by a 1000 by 1000 grid of cells. Each cell manages 20 state variables and the simulation runs for 500 years at 1-week time steps. Assuming all variables are represented with 8-byte floating point values, the entire simulation could generate 416 terrabytes of output (1000*1000*20*500*52*8). A data storage submodel would act like a visualization model that probes the simulation for the state of the system. But, instead of graphically rendering the output, it stores selected portions of the simulation in files for later statistical analysis.

Post Analysis

The data storage submodels capture data for later analysis. I-STEMS will rely on available data analysis and display software for these analyses, including statistical packages, geographical information and image processing systems, and standard graphics depiction tools including translators and movie viewers.

7 Programmer View

Audience

From the programmer's viewpoint, I-STEMS gets into complex technical decisions regarding programming languages, inter-process communication, parallel and distributed processing, object development, and object encapsulation of legacy software. Perhaps the greatest challenge is the choice of the software building-blocks used to develop a large complex system like I-STEMS. Hardware and software environments are still changing very fast. Although it is imperative that system development environments be chosen, emerging technologies can rapidly age such choices. Hence, choices must be made with respect to the anticipated release schedule for the software under development. As this schedule has not been established, this document will only suggest potential choices and focus on the requirements of the system from the perspective of a programmer.

System Design Philosophy

I-STEMS is intended to be a general-purpose, dynamic, spatial, ecological modeling system. As such it must be highly modular, adaptable, and interesting to a broad audience of research institutions and programmers. It will not be financially possible for a single organization to design and develop the entire capability. Therefore, modularity is an absolutely essential requirement.

I-STEMS must allow for the adoption and adaptation of existing simulation software. As described above, I-STEMS seeks to address the need for land simulation models to communicate with one another. While it may be seductive to imagine the design and development of all new software that can make use of the latest advances in computer hardware and software products and theory, it is essential that I-STEMS developers focus limited I-STEMS resources on techniques that will use existing software. At the expense of simulation efficiency, this avoids the cost of reproducing, debugging, and supporting replacement software and allows experts in the modeling and simulation community the opportunity to participate in I-STEMS with minimal investments.

System Overview

There must be a heart to I-STEMS that provides the glue or focus for the system. This will be the underlying submodel intercommunication standards and language. To some extent it will also be a core set of system viewers and controllers. An overview of the I-STEMS design is captured in Figure 5. The three large boxes represent different, potentially heterogeneous, computers. Within each computer are a number of different processes connected by data exchange busses. All of the software components operating together represent a simulation model. The ovals represent submodels that a model developer has assembled from a library of modules to address the modeling needs of a land manager.

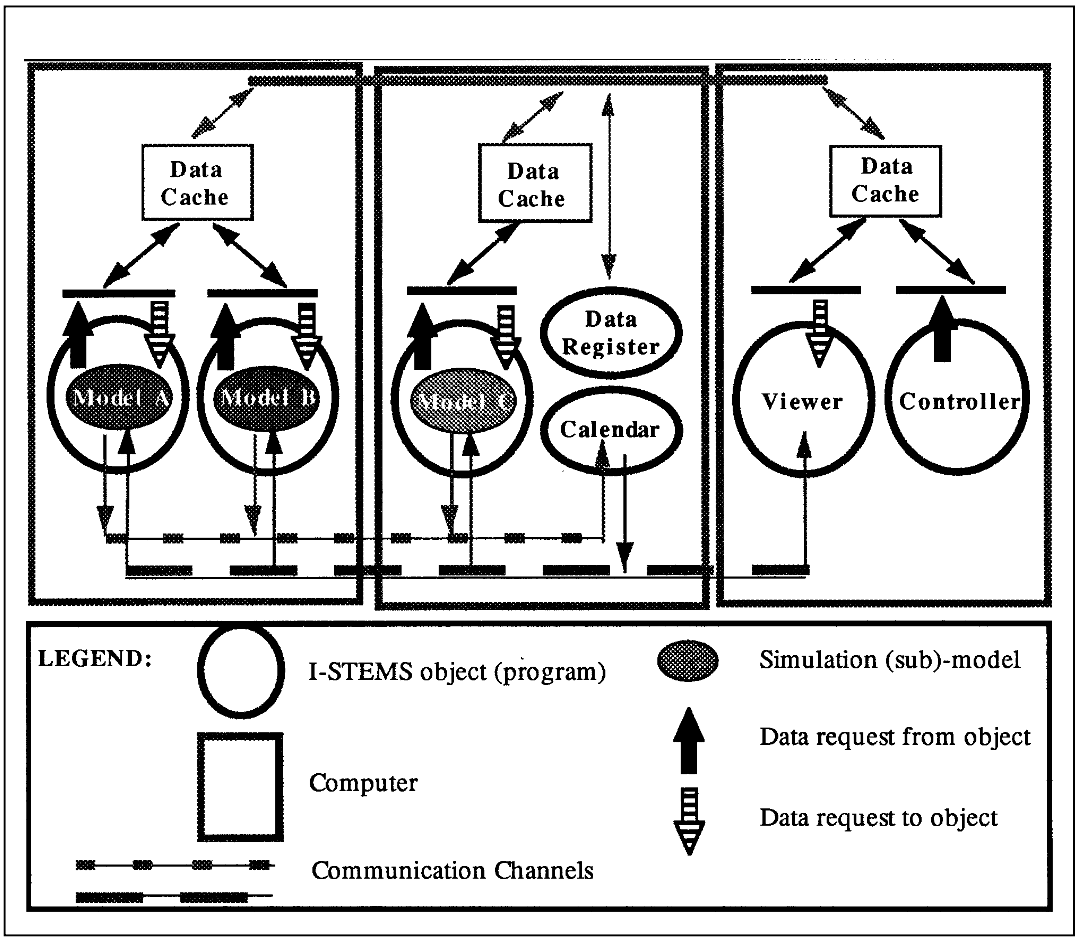


Figure 5. Overview.

The solid ovals in Figure 5 represent three different submodels that may be legacy software models that may have originally operated as standalone programs. Each of these submodels is encapsulated as an I-STEMS object (represented by unfilled ovals). Each object can run as a separate process on a particular computer (or network of computers, or multiple CPUs within a single computer). The encapsulation provides standard communication channels for requesting information from other submodels (thick black arrows pointing up) and for responding to such requests initiated from other objects (thick gray arrows pointing down). Notice the variety of I-STEMS objects (unfilled ovals). Each communicates with other system objects with a standard set of protocols over a limited number of "channels."

Three "channels" are suggested in Figure 5. The topmost bar, above the "data cache" objects in the diagram, represents exchange of data between objects as mediated by the data caches. The bottommost bars provide communication between the timekeeper and the systems model objects.

Simulation Timekeeper

I-STEMS submodel objects must run in synchronous simulation time. It is always presumed that the various submodels in an I-STEMS simulation require current information from other submodels. Hence, it is important that each submodel remain synchronous with a central clock that keeps simulation time. This is the "calendar" object in Figure 5. As a calendar, it accepts requests from the simulation model objects. These objects essentially schedule themselves for updates or actions that they must perform at a later time. For example, a hydrologic simulation object might schedule itself for a full update at a particular time. When the calendar reaches that time, it alerts the scheduled object.

Subsystem Encapsulation

Each modeling capability added to I-STEMS will conform to strict appearance standards. A design requirement allows direct communication of a encapsulated model component with only the locally running data cache. Hence, any information provided by other model objects running within the same simulation must be provided in standard formats in response to standard requests. This approach makes it possible to add new simulation model components to I-STEMS without having to reprogram existing components.

The steps required for subsystem encapsulation of an existing standalone simulation model involve:

- Separation of the model from the data and user interface
- Connection of the model to standard I-STEMS encapsulation specifications
- Development of a simulation to test the new model.

Before developing these steps, it must be reaffirmed that the best group or person to perform the above steps is the original author of the standalone simulation model. Doing so typically minimizes the development costs. It also helps ensure a link to future developments on the original system. Finally, it minimizes later debugging problems and greatly decreases debugging costs. The core I-STEMS development team will be well advised to contract out most design and development of I-STEMS model components. The core team should focus on the development, enhancement, and maintenance of the core system components described elsewhere in this chapter.

The first step in the list above is the separation of the actual modeling code from any data and user interface. I-STEMS model objects interface with the rest of the world through communications with its associated data cache. All communications between any given model and data sources, other models, and humans is accomplished through the data cache. Therefore, the actual model must be isolated from all of its communications. Requests for data and the ability to respond to data calls must then be connected to standard I-STEMS model encapsulation routines.

Encapsulation routines will provide a variety of functionalities. These are described below. The actual specification of how these capabilities will be realized is not part of this document. A variety of implementation details are possible and will be the responsibility of an I-STEMS development team. The required functionalities are covered here.

1. *Register model with local data cache.* Encapsulated I-STEMS submodels will run as separate processes. At startup time, the submodel will be provided with its local cache and with how it communicates with the master clock. At startup, the model is required to register itself with the local cache.

2. *Identify data inputs that will be required at startup*. Typically, at startup the submodel identifies to the local cache four types of information. The first type is the information that will be required to initialize the model. The second will be the data that will be required during a simulation. The third is information that this model can provide at startup. And fourth is data that the model can provide during a simulation.

3. *Monitor simulation clock*. Another startup action is to establish communication with the system simulation clock. This is followed by monitoring the clock through the simulation run(s).

4. *Register times with clock (announce and wait)*. In addition to monitoring the simulation clock, each simulation model will be optionally able to communicate two types of messages to the clock. First, the model can tell the clock to transmit the time at a particular simulation time, and then wait for a go-ahead message from the clock. Second, a go-ahead message can be sent. Events can be scheduled by telling the clock to transmit the time when the scheduled time is met. Those model events that must be completed synchronously return a go-ahead upon completion of the event. Asynchronous events will be completed after first sending the go-ahead to the clock. Semi-synchronous events can send a second schedule time followed by the go-ahead. This second time represents the time when the semi-synchronous event must be completed. The submodel sends a go-ahead when the event in progress is completed.

5. *Receive initialization data*. The beginning of a simulation is a unique event. The state of the system being simulated must be loaded into the system. The operation of a simulation involves three basic phases. First, the I-STEMS core simulation software is started. This involves the main system clock and data register. This runs on a single machine in a network and is associated with set communication channels. Data caches are also started on each machine participating in the simulation. These establish communications with each other and with the data register. Second, the various models, viewers, and controllers are initialized. They communicate their data requirements and offerings to their associated data caches that share this information with the data register. Third (and finally), after all the data requirements are met, a simulation can be started. This involves moving all of the initialization data from data-providing objects and then starting the calendar.

6. *Request and receive data.* During simulations, data will be moved back and forth between the different models. Each model will request and receive data.

7. *Receive and respond to data requests.* Data requests will make their way to the models that supply the data. Each model must accept and respond to these requests.

8. *Reset.* Each model must respond appropriately to a reset signal. At this signal, each will reinitialize itself so that it is in the identical state it was in at first start-up.

As described above, data is being moved between the different models running as separate programs. A standard set of data formats will be used for transmission of this system state information. These will include:

- A bounded raster of data (integer, floating point, null)
- State at a given point (a particular piece of information at a particular point)
- State within a radius at a given point (returns the average value for a given piece of information)
- Others.

All data will also be associated with units. Data requests and responses must match units. Unit conversion will be accomplished, when needed, in the process of moving data from the data cache to the requesting sub-model.

Data Cache Objects and the Data Register

I-STEMS submodels do not view the external world as a set of objects, but rather as a set of available information. Each object operates with the “belief” that it is the center of the known universe and is surrounded with information that it can probe. It also responds to information requests, but is unaware of where the requests originate. That external world of information and information requests from the world is mediated by a data cache object. One data cache object is running on each participating computer and is known to each I-STEMS object running on that computer. Because I-STEMS objects do not see other I-STEMS objects directly, the complexity in communicating information is minimized. This reduces the size of the API (Application Programmer Interfaces), minimizing the learning required for new I-STEMS programmers and also minimizing the effort required to create, test, and validate new I-STEMS objects.

The data cache running on any one machine communicates with all other data caches running on all other machines cooperating in a particular I-STEMS simulation. Each cache provides the following functionalities:

1. Maintain information about data managed by its I-STEMS objects. This includes the format of the data (raster map, vector map, parameter, value, error measure, and lifetime of the data). This information can be fetched from other data caches when needed and stored locally. It may be provided to its objects when requested. If shared memory is available, it might instead provide the memory location of the data to the objects.

2. Map of where each possible data requirement for its I-STEMS objects can be located.

The data cache also communicates with an I-STEMS simulation data register. This is associated with a set of “master controls” and manages the location and type of all available data that will be generated by a set of I-STEMS objects. Data caches query this register to find out where each required data type can be found on the network of I-STEMS objects.

Viewers and Controllers

On the right side of Figure 5 are represented a viewer and a controller. “Subsystem Encapsulation” (p 53), the process of converting an existing standalone model into an I-STEMS submodel first involves isolating the model from the data and user interface. In I-STEMS, the user interface is replaced with special submodels that interact with computer peripherals including monitors, keyboards, and data storage devices. Because these communicate with other components of a complete simulation model via the data caches, the submodels themselves know nothing of the user interface and visualization details.

It is expected that a number of viewers and controllers will be developed to allow appropriate user feedback and input. Some of those expected might be described as follows:

1. *Strip-chart viewer*—The user will associate one or more streams of data acquired by regularly querying information in submodels. An interactive version will allow a user to peruse the different data available from the submodels and dynamically select the data they wish to track.

2. *Error message monitor* — Submodels may generate error messages that can be captured and displayed.

3. *Map*—Mapped information can be dynamically extracted from a submodel and displayed. Various levels of cartographic information like grids, overlays, labels, and coordinates may be optionally displayed.

4. *Movie*—A variant of the 2-D map, this viewer will allow the history of the simulation to be viewed up to the current simulation time.

5. *State monitor*—Some submodels will simulate the state of some discrete landscape entity. The internal state and external environment of these entities may be accessed and viewed.

6. *Capture*—Each of the above monitors may optionally provide the ability to save the data or the images to files for post-processing. Additionally, some viewers will need to do little more than allow the user to select available data for dynamic storage without rendering the data during a simulation.

7. *Person-in-the-loop*—This will be a controller that allows a person to manipulate or adjust the behavior of some landscape entity with a controller interface. For example, the behavior of an animal might be provided dynamically by a scientist familiar with that animal. Some submodels might allow a controller to adjust internal parameters, thereby allowing a population to artificially recover, the weather to change, an infestation to begin, or zoning legislation to change. It is likely that some full simulation models will be developed that do simulations based only on the dynamic input of a number of users. This type of a gaming environment can become very important in exploring alternative approaches to land management.

There will be other viewers and controllers, and there will be a number of competing versions. I-STEMS must be an open system that can be adopted by a wide variety of research labs.

Implementation Approaches

Implementation of an integrated spatio-temporal ecological modeling system can be accomplished with hardware and software technologies available now in the late 1990s. The biggest challenge is amassing sufficient interest in one (or a consortium) of organizations to pull together the first prototype. This is an organizational and leadership challenge. We will first look at alternative technical approaches and then explore potential management approaches.

Before developing any software, it is critical that management target the intended audience. Two critical questions must be answered:

1. *Who will be the intended, or target, user community?* Planners? Scientists? Regional offices with large staffs? Local offices staffed with one or two people? City planning offices? Agricultural planning offices?

2. *During what years is the system expected to be viable?* This question is easily overlooked. A system created to be useful for a single user to complete a study next year is much different than a system designed to be viable over a decade or more.

It is recommended that the I-STEMS system be developed for small to medium land management offices and that the system be viable between 1999 and 2010. The time-line is targeted to be:

1996-1997: Core system design and development

1997: Submodel encapsulation Application Programmer Interface (API) prototype complete

1997: Publication of API design

1997: Publication of Draft User’s Manual

1997-1999: Encapsulation of existing models to submodels

1997-1998: Basic Controllers and Viewers

1997-1999: Development of sample landscape simulation models

1998: Alpha release

1999: Beta and Final Version 1.0 release

Associated with this technical timeline are management requirements that are discussed at the end of this section. Note the major milestones. The publication of the User's Manual and API design occurs relatively early in the development process; these materials are crucial to the development programmers.

During this time period, hardware and software environments are to be appropriately targeted. It is difficult for hardware manufacturers to look 4 years into the future. Projecting current trends should be sufficient for this purpose. I-STEMS, as described in this chapter, presumes that any office using it will have Internet access, may have a number of heterogeneous machines, and that these machines may have multiple processors. At a minimum, I-STEMS adopted for a regional office might require the following machine:

CPU: 1 200Mhz processor

Disk: 4 Gbyte

Memory: 64 Mbyte

Peripherals: Monitor, keyboard, color printer

In 1996 such a machine could be purchased in the $10K range. This capability already exists at many of the small to medium offices. A network of several of these machines all served by a common data base is typical. The hardware that exists at many offices is already sufficient to run I-STEMS type software. It is safe to anticipate that over the next 5 years, the cost of computer hardware will continue to decrease as capability increases. There are not likely to be any unanticipated fundamental revolutions in hardware that will affect I-STEMS. New devices will do the same jobs faster and cheaper.

Software is more difficult to predict. In the mid-90s the PC-compatible has been the dominant machine in the workplace. A number of different operating systems exist with Windows-95 being the standard on PC class machines, UNIX on workstations, and MacOS on Macintosh platforms. Windows-95, MacOS, and similar single-user operating systems are inadequate for supporting I-STEMS. Windows-NT shares characteristics with UNIX and Windows-95. It runs on a number of different machines. In particular, it runs on any machine that supports Windows-95. In addition, it is multi-user, multi-processor, multi-threaded, and supports many of the capabilities of UNIX. I-STEMS development should target both Windows-NT and UNIX for the foreseeable future.

The question of the selection of a computer language for I-STEMS is very difficult. It is anticipated that programming languages will continually be created and improved. Because the languages of existing simulation software that I-STEMS will adopt are numerous, and because potential I-STEMS submodel developers will choose any of a number of different languages, it is a design goal that multiple languages be supported. However, the core language used to provide that support invisibly to the programmers should be a single language. Candidates include C, C++, Java, and Objective C. It is recommended at this point that the choice of language be made by the I-STEMS development team once it is assembled. That team must evaluate the technical capabilities, market availability, and preferences of participating organizations in its consideration of language.

Choice of language must also reflect the availability of software libraries that already exist and are determined to be technically useful and transportable, and will be supported through the life of I-STEMS. One type of library that may prove indispensable is an implementation of the Common Object Request Broker Architecture (CORBA). This is a specification that allows objects, running as different processes, to interact with each other; a key design feature of I-STEMS. CORBA implementations are becoming available and support different languages and different systems. Some are supported. Some are commercial.

Regardless of the specific decisions made with respect to operating system, hardware platforms, libraries, and programming languages, it is imperative that the I-STEMS design and development be as modular as possible. Adopting an object-oriented approach is helpful in forcing modularity. Modularity is expensive during the design and development phase, but is crucial in extending the useful life of the product.

Management of an I-STEMS development project will face several critical challenges. Funding and collaboration are the most important and are inseparable. Collaboration is important partly because it provides a broader funding base. More importantly, however, the goal of I-STEMS is to integrate the best models of scientists into excellent models for making management decisions. Because the models on which I-STEMS integrated models will be based are the end result of significant funding and intellectual effort, it is important to have the original development teams involved in the creation of I-STEMS simulation modules. I-STEMS management will be advised to hold workshops and clinics early and often to maximize the buy-in from the broadest audience possible.

Although there should be significant participation from a large community, I-STEMS must be associated with a small, talented, and dedicated programming staff. This staff must remain consistently funded and remain relatively unchanged throughout the critical first years of development.

8 Review of Existing Systems

A number of different systems currently exist or are under development at different research organizations

MMS—The Modular Modeling System

Developer

Dr. George Leavesly, U.S. Geological Survey

Description

The Modular Modeling System is best introduced with a paragraph from *The Modular Modeling System — MMS: User's Manual* (Leavesley, Restrepo, et al. 1995):

> The Modular Modeling System is an integrated system of computer software developed to (1) provide the research and operational framework needed to enhance development, testing, and evaluation of physical-process algorithms; (2) facilitate integration of user selected algorithms into operational physical-process models; and (3) provide a common framework in which to apply historic or new models and analyze their results. MMS uses a module library that contains modules for simulating a variety of water, energy, chemical, and biological processes. A model is created by selectively coupling appropriate modules from the library to create a suitable model for a desired application. When existing modules do not provide appropriate process algorithms, new modules can be developed.

Current information on MMS can be accessed on the Internet (Leavesley 1996).

Review

MMS approaches integrated modeling and simulation at the subroutine level. It provides a common data exchange capability for sharing data between subroutines compiled into a single program running on a single computer. The project has attracted funding from a wide variety of collaborators including several European organizations. It has been led with a consistent vision that has allowed it to continue to develop over nearly a decade.

Fundamentally, MMS differs from the I-STEMS description in one key regard. I-STEMS integrates programs while MMS integrates subroutines.

DIAS—Dynamic Interactive Architecture System

Developer

John Christiansen, DIS Division, Argonne National Laboratories

Description

DIAS is a software environment that facilitates run-time interactions between simulation models. Models captured as standalone programs are treated as software objects that can run anywhere on a network of computers. DIAS provides the capabilities that provide intercommunication between the different objects.

Review

DIAS is a proven and working system that provides virtually all of the I-STEMS specifications. A release of DIAS was scheduled for the Spring of 1997. Funding to support DIAS has come from a wide range of sponsors including the Joint Chiefs of Staff/J-8 for use in developing a prototype terrain reasoning and synthetic terrain generation system, and by the USAF Air Weather Service as the software framework for a multi-disciplinary environmental modeling effort in support of theater-level mesoscale weather forecasting.

HLA—High Level Architecture and RTI–Run-time Infrastructure

Description

The Defense Modeling and Simulation Office within the Department of Defense has developed a software architecture with the help of Lincoln Laboratories and MITRE Corporation (DMSO 1996). Called the High Level Architecture (HLA), it provides a specification for developing software simulations that can interact with one another. Such simulations are called federates and can run across a network of computers. They interact with one another via the Run-time Infrastructure (RTI). As operating systems residing on individual computers provide services to programs, the RTI similarly provides services to federates running on a network. RTI can be thought of as a distributed operating system that provides:

- Federation Management
- Declaration Management
- Object Management
- Ownership Management
- Time Management
- Data Distribution Management.

Review

HLA and RTI provide for every I-STEMS requirement. HLA baseline architecture design requirements were published in September 1996. These documents can be found with a WWW browser at: http://www.dmso.mil/ projects/hla/. All DOD modeling and simulation efforts are expected to become compliant with the HLA specifications. It is not only highly recommended, but required that I-STEMS development efforts comply with HLA and RTI specifications.

ALSP—Aggregate Level Simulation Protocol

Description

The Aggregate Level Simulation Protocol (ALSP) is currently used by the National Simulation Center (NSC) (http://www-nsc.army.mil/) to support real-time battlefield simulators. Funded by DOD, ALSP is a product of the MITRE Corporation. The government contract manager for ALSP is Dr. Connie Fischer at U.S. Army Simulation, Training, and Instrumentation Command (STRICOM). ALSP provides a forum for supporting Distributed Interactive Simulation (DIS) software for battlefield simulation.

Review

ALSP is scheduled for replacement by new software that complies with HLA. It will be maintained in the interim to support currently operating battlefield simulators.

9 Summary

This document provides a working design for the Integrated Spatio-Temporal Ecological Modeling System (I-STEMS). It is designed as a geographic modeling system (GMS) that could be operational during the years 2000 to 2015. I-STEMS is designed to meet the needs of natural resource managers for anticipating the state of a landscape over time based on scheduled land-use patterns, historic and current records, and rules that capture the interaction of the landscape and human activities and responses.

I-STEMS provides the simulation environment within which to build land management decision support systems for Army land managers. This design document focuses on I-STEMS itself and is therefore of most interest to potential I-STEMS programmers and decision support system simulation model developers.

Simulation modeling has become increasingly popular and effective in many different fields. It is particularly useful for complex systems composed of well understood components. In recent years, simulation modeling has come to land management, resulting in the development of a number of different products. Each focuses on some aspect of the environment. Products available or becoming available include stormwater runoff, groundwater movement, training impacts, vegetation recovery and succession, soil erosion, air pollution, and species specific models. An emerging problem is that these different models provide different results because each simulates only a portion of an interacting ecological system. It is often implicitly recognized that the whole system must be represented in such models, and often attempts are made to do so. The dynamics within each model, however, focus on the knowledge of only a single discipline.

Efforts are being made in the research community to integrate disparate simulation systems. There are perhaps three distinct approaches. The first is to have the different systems simply run in serial, sharing information with one another through standard data files and formats. The second is to link disparate models into a single computer program to provide faster simulations that can dynamically exchange information between the simulation components.

This approach has the drawback of creating very large and unwieldy programs authored by a committee. Object-oriented programming approaches are allowing software contributors more autonomy while ensuring end-product interaction. I-STEMS describes a third approach that recommends the development of existing simulations into standalone programs that, at run-time, communicate and interact with other associated simulations.

A number of different development efforts funded by the battlefield simulation community are effectively addressing the I-STEMS approach. It is recommended that the land management community collaborate with efforts well underway.

Bibliography

Allen, T.F.H., and T.B. Starr, *Hierarchy* (University of Chicago Press, 1982).

Andrewartha, H.G., and L.C. Birch, *The Distribution and Abundance of Animals* (University of Chicago Press, 1954).

Ball, G.L., and R. Gimblett, "Spatial Dynamic Emergent Hierarchies Simulation and Assessment System," *Ecological Modeling*, vol 62 (1992), pp 107-121.

Botkin, D.B., ed., *Life and Death in a Forest: The Computer as an Aid to Understanding, Ecosystem Modeling in Theory and Practice, An Introduction With Case Studies* (Wiley and Sons, 1977).

Botkin, D.B., J.F. Janak, et al., "Some Ecological Consequences of a Computer Model of Forest Growth," *Journal of Ecology*, vol 60 (1972), pp 849-872.

Buckley, D.J., M. Coughenour, et al., "The Ecosystem Management Model Project: Integrating Ecosystem Simulation Modeling and ARC/INFO in the Canadian Parks Service," *Second International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling* (Breckenridge National Center for Geographic Information and Analysis, Colorado, 1993).

Caswell, H., "Community Structure: A Neutral Model Analysis," *Ecological Monographs*, vol 46 (1976), pp 327-354.

Daniels, R.E., and H.E. Burkhart, "An Integrated System of Forest Stand Models," *Forest Ecology and Management*, vol 23 (1988), pp 159-177.

Delcourt, H.R., and P.A. Delcourt, "Quaternary Landscape Ecology: Relevant Scales in Space and Time," *Landscape Ecology*, vol 2 (1988), pp 23-44.

de Castro, F., "Computer Simulation of the Dynamics of a Dune System," *Ecological Modelling*, vol 78 (1995), pp 205-217.

Denning, P.J., "The Science of Computing; Modeling Reality," *American Scientist*, vol 78 (1990), pp 495-498.

DMSO, *DOD High Level Architecture*, http://www.dmso.mil/projects/hla (1996).

Dreyfus, S., "How Reliable Are Computer Models of Socioeconomic Systems?", *ACM Conference on Critical Issues in Computing* (ACM Press, 1990).

Forrester, J., "Models and the Real World," *ACM Conference on Critical Issues in Computing* (ACM Press, 1990).

Fulton, M.R., "A Computationally Efficient Forest Succession Model: Design and Initial Tests," *Forest Ecology and Management*, vol 42 (1991), pp 23-34.

Gardner, R.H., B.T. Milne, et al., "Neutral Models for the Analysis of Broad-Scale Landscape Pattern," *Landscape Ecology*, vol 1 (1987), pp 19-28.

Gardner, R.H., R.V. O'Neill, et al., "Ecological Implications of Landscape Fragmentation," in *Humans as Components of Ecosystems: The Ecology of Subtle Human Effects and Populated Areas* (S.T.A. Pickett and M.J. McDonnell, New York, NY, Springer-Verlag, 1993).

Gardner, R.H., M.G. Turner, et al., "Simulation of the Scale-Dependent Effects of Landscape Boundaries on Species Persistence and Dispersal," *The Role of Landscape Boundaries in the Management and Restoration of Changing Environments* (P.G.R. & R.J. Naiman, New York, Chapman and Hal**,** 1991), pp 76-89.

Gilpin, M.E., "Extinction of Finite Metapopulations in Correlated Environments," in B. Shorrocks and I. R. Swingland, *Living in a Patchy Environment* (Oxford University Press, 1990), pp 177-186.

Hannon, B., "The Structure of Ecosystems," *Journal of Theoretical Biology*, vol 41 (1973), pp 535-546.

Hannon, B. "Ecosystem Flow Analysis," *Canadian Bulletin of Fisheries and Aquatic Sciences*, vol 213 (1985), pp 97-118.

Hanski, I., "Single-Species Spatial Dynamics May Contribute to Long-Term Rarity and Commonness," *Ecology*, vol 66 (1985), pp 335-343.

Hanski, I., and M. Gilpin, "Metapopulation Dynamics: Brief History and Conceptual Domain," *Biological Journal of the Linnean Society*, vol 42 (1991), pp 3-16.

Horn, H.S., and R.H. MacArthur, "Competition Among Fugitive Species in a Harlequin Environment," *Ecology*, vol 53 (1972), pp 749-752.

Johnson, A.R., "Spatiotemporal Hierarchies in Ecological Theory and Modeling," *Second International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling* (Breckenridge, CO, 1993).

Kline, S.J., *A Numerical Measure for the Complexity of Systems: The Concept and Some Implications*, (Stanford University, Department of Mechanical Engineering, 1990).

Klijn, F., and H.A. Udo de Haes, "A Hierarchical Approach to Ecosystems and Its Implications for Ecological Land Classification," *Landscape Ecology*, vol 9, no. 2 (1994), pp 89-104.

Leavesley, G., *The Modular Modeling System* (U.S. Geological Survey, Denver, CO, 1996).

Leavesley, G.H., P.J. Restrepo, et al., *The Modular Modeling System—MMS: User's Manual* (U.S. Geological Survey, Denver, CO, 1995).

Levin, S.A., "Ecology in Theory and Application," in S.A. Levin, R.G. Hallam, and L. J. Gross, *Applied Mathematical Ecology* (Berlin, Springer-Verlag, 1989), pp 3-8.

Levins, R., and D. Culver, "Regional Coexistence of Species and Competition Between Rare Species," *Proceedings of the National Academy of Sciences*, vol 68 (1971), pp 1246-1248.

Liu, J., "ECOLECON: An ECOLogical-ECONomic Model for Species Conservation in Complex Forest Landscapes," *Ecological Modelling*, vol 70 (1993), pp 63-87.

Loh, D.K., and Y.-T.C. Hsieh, "Incorporating Rule-Based Reasoning in the Spatial Modeling of Succession in a Savanna Landscape," *Al Applications*, vol 9, no. 1 (1995), pp 29-40.

Loucks, O., T.W. Doyle, et al., "Concepts, Theory and Models of Forest Succession"

MacArthur, R.H., and E.O. Wilson, *The Theory of Island Biogeography* (Princeton University Press, 1967).

Maxwell, T., *Distributed Modular Spatial Ecosystem Modeling*, http://kabir.umd.edu/SMP/MVD/CO.html (University of Maryland, 1995).

Maxwell, T., and R. Costanza, "Spatial Ecosystem Modeling in a Distributed Computational Environment," in J. van den Berg and J. van der Straaten, *Concepts, Methods, and Policy for Sustainable Development* (Island Press, 1993).

May, R.M., "The Search for Patterns in the Balance of Nature: Advances and Retreats," *Ecology*, vol 67, no. 5 (1986), pp 1115-1126.

O'Neill, R.V., D.L. DeAngelis, et al., *A Hierarchical Concept of Ecosystems* (Princeton University Press, 1986).

O'Neill, R.V., R.H. Gardner, et al., "A Hierarchical Neutral Model for Landscape Analysis," *Landscape Ecology*, vol 7, no. 1 (1992), pp 55-61.

O'Neill, R.V., A.R. Johnson, et al., "A Hierarchical Framework for the Analysis of Scale," *Landscape Ecology*, vol 3 (1989), pp 193-206.

Perestrello de Vasconcelos, M.J., B.P. Zeigler, et al., "Modeling Multi-Scale Spatial Ecological Processes Under the Discrete Event Systems Paradigm," *Landscape Ecology*, vol 8, No. 4 (1993), pp 273-286.

Saghafian, B., "Implementation of a Distributed Hydrological Model Within Geographical Resources Analysis Support System (GRASS)," *Second International Conference/Workshop on Integrating Geographic Information Systems and Environmental Modeling* (Breckenridge, CO, 1993).

Sklar, F.H., R. Costanza, et al., “Dynamic Spatial Simulation Modeling of Coastal Wetland Habitat Succession,” *Ecological Modelling*, vol 29 (1985), pp 261-281.

Slatkin, M., “Competition and Regional Coexistence,” *Ecology*, vol 55 (1974), pp 128-134.

Starfield, A.M., and A.L. Bleloch, *Building Models for Conservation and Wildlife Management* (Collier Macmillan, London, 1986).

Stommel, H., “Varieties of Oceanographic Experience,” *Science*, vol 139 (1963), pp 572-576.

Turner, M.G., “Landscape Ecology: The Effect of Pattern on Process,” *Annual Review of Ecology and Systematics*, vol 20 (1989), pp 171-197.

Turner, M.G., R.H. Gardner, et al., “Predicting the Spread of Disturbance Across Heterogeneous Landscapes,” *Oikos*, vol 55 (1989), pp 121-129.

Urban, D.L., R.V. O’Neill, et al., “Landscape Ecology, A Hierarchical Perspective,” *BioScience*, vol 37 (1987), pp 119-127.

Vasconcelos, M.J., B.P. Ziegler, et al., “Modeling Multi-Scale Spatial Ecological Processes Under the Discrete Event Systems Paradigm,” *Landscape Ecology*, vol 8, no. 4 (1993), pp 273-286.

Verboom, J., and K. Lankester, “Linking Local and Regional Dynamics in Stochastic Metapopulation Models,” *Biological Journal of the Linnean Society*, vol 42 (1991), pp 39-55.

Vieux, B.E., and J. Westervelt, “Finite Element Modeling of Storm Water Runoff Using GRASS GIS,” *Computing in Civil Engineering and Geographic Information Systems Symposium* (American Society of Civil Engineers, Dallas, TX, 1992).

Wiens, J.A., J.F. Addicott, et al., “Overview: The Importance of Spatial and Temporal Scale in Ecological Investigations,” in J. Diamond and T.J. Case, *Community Ecology* (Harper and Row New York, NY, 1985), pp 169-193.

Wissel, C., “Aims and Limits of Ecological Modelling Exemplified by Island Theory,” *Ecological Modelling*, vol 63 (1992), pp 1-12.

Wittmann, J., “The SIMPLEX II Simulation System To Solve Problems in Environmental Protection,” *Ecological Modelling*, vol 75 (1994), pp 563-573.

Wu, J., “Balance of Nature and Environmental Protection: a Paradigm Shift,” *Fourth International Conference of Asia Experts—China’s Environment: Meeting Local and Global Challenges* (Portland State University, Portland, OR, 1992).

Wu, J., and O.L. Loucks, “Balance-of-Nature and Modern Ecological Theory: A Shift in Ecological Thinking,” *Development and Trends in Modern Ecology* (Sino-ECO, Hefei University Science and Technology Press, 1991).

Wu, J., J.L. Vankat, et al., "Effects of Patch Connectivity and Arrangement on Animal Metapopulation Dynamics: A Simulation Study," *Ecological Modelling*, vol 65 (1993), pp 221-254.

USACERL DISTRIBUTION

Chief of Engineers
ATTN: CEHEC-IM-LH (2)
ATTN: CEHEC-IM-LP (2)
ATTN: CECC-R
ATTN: CERD-L
ATTN: CERD-M (2)

Defense Tech Info Center 22304
ATTN: DTIC-O (2)

10
5/98