reprinted from

IEEE COMPUTATIONAL SCIENCE & ENGINEERING

A Modular System for Scalable Ecological Modeling

Ray Ford, Steven Running, and Ramakrishna Nemani
University of Montana

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE COMPUTER SOCIETY

EARTH & ENVIRONMENTAL SCIENCES

A Modular System for Scalable Ecological Modeling

Ray Ford, Steven Running, and Ramakrishna Nemani
University of Montana

Ad-hoc growth of a successful modeling system isn't always smooth. As this system was upgraded to simulate the ecology of ever-larger regions, organizing the data and tools became as crucial as improving the science.

THE POWER OF MODELS THAT CAN DESCRIBE AND predict terrestrial ecosystem behavior has advanced dramatically over the last two decades, driven by major improvements in process-level modeling techniques, computing technology, and the availability of diverse data. Success has bred discontent: researchers once satisfied with studying the ecological processes of single plants or small groups of plants now envision models describing regions, continents, or the entire globe. Indeed, scientists ultimately want to model the very large scale terrestrial/atmospheric exchange processes that are crucial to "grand challenge" problems in global ecology.

As the scale and sophistication of ecosystem modeling efforts increase, however, the modeling system must become a more comprehensive information system. Much of the scientific information from simulations exists only in electronic form, embodied in an ever-growing collection of model components, input data sets, and derived (output) data sets. If we cannot effectively access this information, we lose its underlying scientific value.

Our team's experience comes from developing and using the Regional Hydro-Ecological Simulation System (Rhessys, often styled RHESSys) over several years. This experience illustrates both the tremendous potential in computer-based ecological models and the problems associated with emerging ecosystem information systems. We first used computer simulation to study small-scale physiological ecology. In that research, in the early 1970s, we explored the significance of field-gathered measure-

1070-9924/94/$4.00 © 1994 IEEE

ments of the transpiration of moisture from a stand of trees into the atmosphere. From this start, modeling scale and diversity of inputs grew to where our system now has major components incorporating topography, mountain climatology, remote sensing technology, and hydrology. The components of Rhessys are point-based models that can describe certain key ecosystem processes at up to regional scales. (Ecology, though, is a broad field and Rhessys does not model all of its aspects; the accompanying sidebar further explains different types of ecosystem models and where ours fits in.)

Rhessys is increasingly burdened with the growing pains common to research software that evolves in an ad hoc manner on the momentum of its own success, rather than according to a comprehensive plan. These scaling problems have long been familiar to information system managers, and are becoming more apparent in computationally intensive science. We now find that we must address information system problems as seriously as ecosystem problems. We are currently working to incorporate Rhessys into a more usable object-oriented Ecosystem Information System. This reorganization will help add structure to our existing efforts, but more important, it is essential if we are to achieve the scaled-up goals we have set for future modeling efforts.

Rhessys: Scientific foundations and matters of scale

The most basic goal of our modeling is to use computer-based simulation to trace key properties in ecosystem productivity—the overall production of biomass—through time. The model relies on several detailed submodels of critical ecosystem processes. A measure of success for such models is their ability to describe all the key processes in a way that accounts for all system inputs and outputs. That is, can they properly "balance the budget" of inputs, outputs, and elements held within the ecosystem? For example, a model might budget the water in an ecosystem by balancing input from rain and snow, water held in the system in soil moisture or snowpack, and output from runoff and evapotranspiration (which includes evaporation from the ground and water surfaces as well as transpiration from plant leaves). Similar budgets for carbon, nitrogen, and other elements can also be computed, and are central to the study of phenomena such as global warming.

These model calculations help us understand how the terrestrial biosphere responds to climatic perturbations such as the volcanic eruption of Mt. Pinatubo in the Philippines in 1991, which caused the trend in atmospheric CO_2 concentration to change for the first time since monitoring began in the 1960s. More practical applications involve computing forest production potential, watershed hydrologic analysis, and wildlife habitat analysis.

Rhessys is designed to map carbon, water, and nutrient fluxes at a variety of spatial scales. These can range from a single hillslope to a 100- by 50-kilometer region.[1-3] Figure 1 schematically shows the four basic classes of input data Rhessys uses: climate, vegetation,

Types of Ecosystem Models

The general term *ecological model* encompasses a wide variety of different types of models, which describe various ecosystem phenomena at various scales. Clearly they all have bearing on each other, since they simulate interrelated natural processes. Examples of macroscale models include

- the land-surface representations used in climatic studies, often called *soil-vegetation-atmosphere transfer* models
- models of the cycling of water, carbon, and nutrients through ecosystems, collectively called *biogeochemistry* models
- models that simulate the germination or birth and the growth and mortality of individual plants or animals in an ecosystem, the interaction among members of a species, and the interaction among different plant and animal species, collectively called *population dynamics* models
- models that simulate the succession of plant species (and accompanying animals) in an ecosystem over time, called *succession* models
- models of the geographical distribution of plant communities and of whole biomes, called *biogeography* models

There is also a great variety of "ecological models" that describe more microscale, individual plant processes, such as photosynthesis, transpiration, or decomposition.

According to the classification above, Rhessys is a macro-level biogeochemistry model. The Forest-BGC subprocess model within Rhessys is a micro-level biogeochemistry model. Kimmins surveys various types of ecosystem models in more depth.[1] Additional details on Rhessys and its components can be found elsewhere.[2,3]

References

1. J. Kimmins, *Forest Ecology*, Macmillan, New York, 1987.
2. R. Nemani et al., "Regional Hydroecological Simulation System: An Illustration of the Integration of Ecosystem Models in a GIS," in *Environmental Modeling with GIS*, M.F. Goodchild, B.O. Parks, and L.T. Steyaert, eds., Oxford Univ. Press, New York, 1993, pp. 296-305.
3. L.L. Pierce et al., "Ecohydrological Changes in the Murray-Darling Basin. III. A Simulation of Regional Hydrological Changes," *J. Applied Ecology*, Vol. 30, 1993, pp. 283-294.

topography, and soils. Most of the raw data inputs to Rhessys come from standard data sets available in electronic form from US government agencies, as indicated by the acronyms listed under Source. Using these inputs, Rhessys components compute and output the derived data sets suggested under Derived Inputs and GIS Preprocessing, each representing an abstraction of an important ecosystem property. Ultimately, these outputs are used as inputs to three major processing components:

- Forest-BGC provides the basic process model of vegetation productivity at the level of an individual stand of trees.
- Topmodel adds a hydrology model that describes soil moisture transmissivity and overland flow of water.
- MT-Clim adds a climatology model that estimates climatic conditions across large regions and highly variable terrain.

Rhessys outputs can be used directly in a number of applications. Once evapotranspiration rates are known, hydrologists can compute stream discharge, a key factor in water management, from precipitation. Net photosynthesis is the primary factor in predicting plant growth.

Forest-BGC (the acronym stands for biogeochemical cycles), the point-based process model core of Rhessys, combines various process-level models to estimate the cycling of carbon, water, and nitrogen through a forest ecosystem (see Figure 2).[4] The model requires daily data for meteorological conditions—maximum and minimum temperatures, dew point, incident solar radiation, and precipitation. It also requires one-time specification of key site and vegetation variables. Forest-BGC calculates biogeochemical values as follows:

- for *water* (H_2O): interception by and evaporation from the forest canopy, transpiration, snowmelt, soil-water content, and outflow from soil;
- for *carbon* (C): photosynthesis, growth respiration, maintenance respiration, allocation, leaf litterfall to forest floor, and decomposition; and
- for *nitrogen* (N): deposition, uptake, litterfall, and mineralization.

As illustrated by the two major subsystems in the schematic, Daily and Yearly, Forest-BGC has a mixed time resolution: hydrologic, photosynthetic, and maintenance respiration

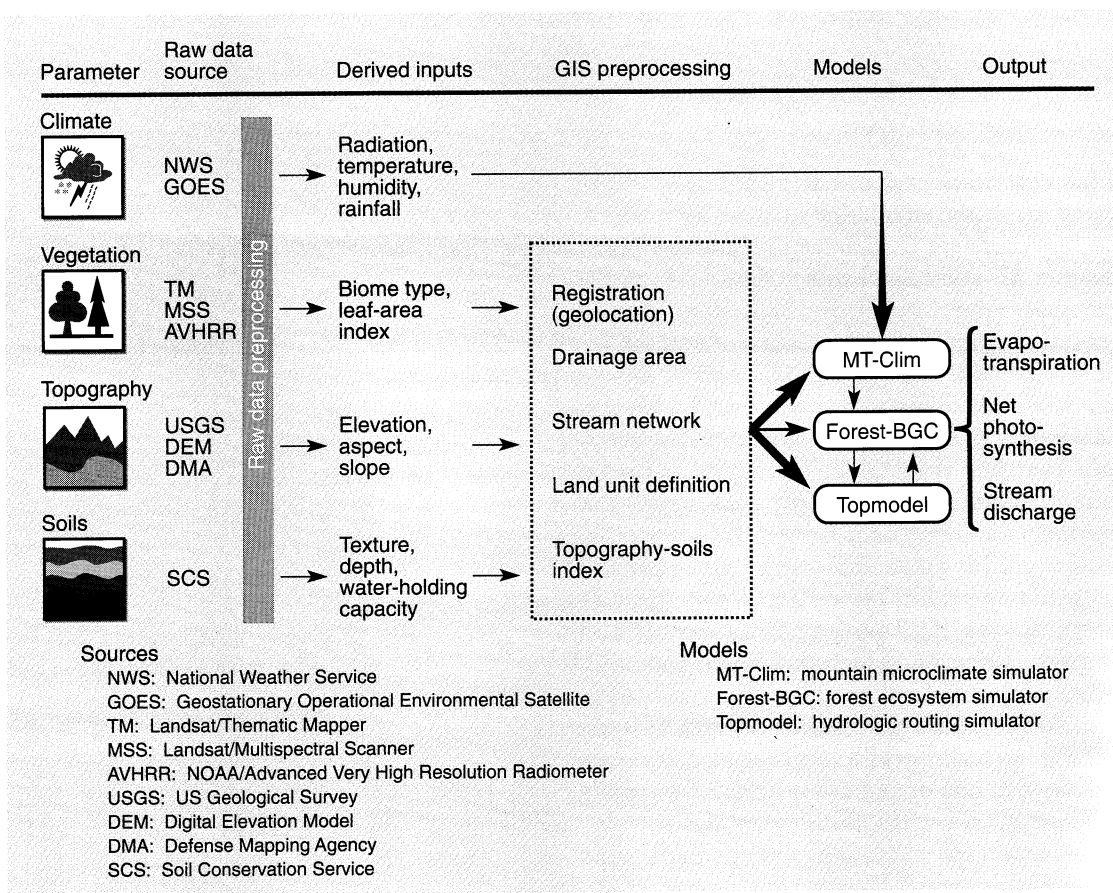


Figure 1. Rhessys, the Regional Hydro-Ecological Simulation System, is shown here in schematic form. Raw data sets from various standard sources are preprocessed and then fed into interacting submodels that compute estimates of ecological functions.

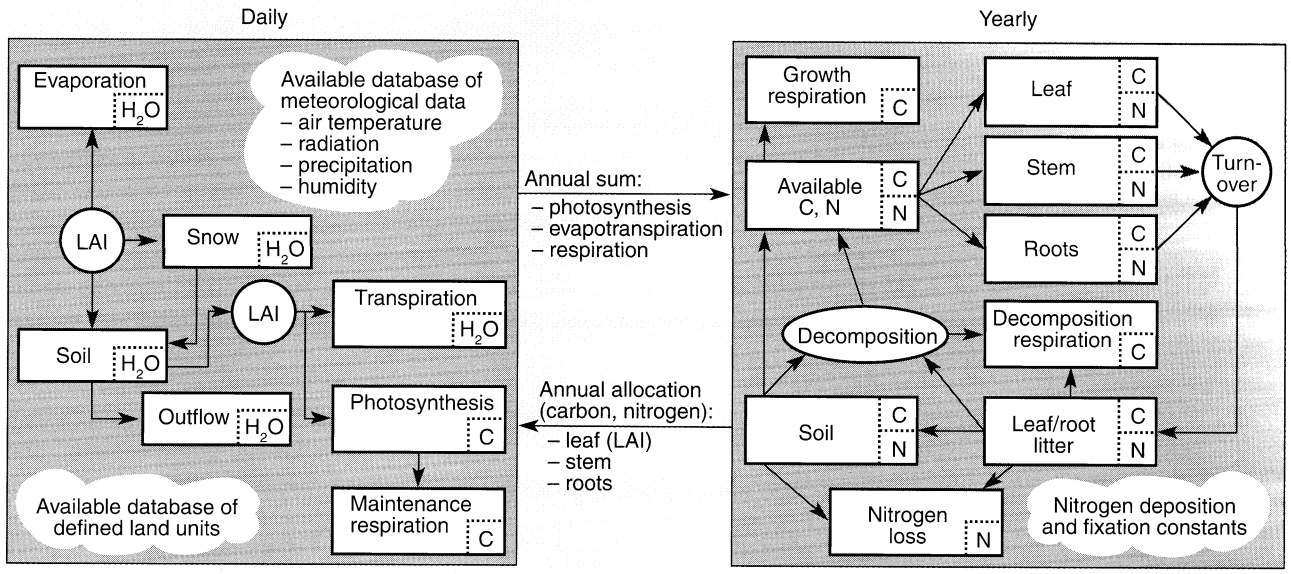


Figure 2. A schematic of Forest-BGC, the core of Rhessys, shows how this module simulates biogeochemical cycles in a forest ecosystem. Forest-BGC works with the other submodels to convert preprocessed data, much of it remotely sensed, into computed values used to measure overall vegetation productivity.

processes are computed daily, while the other carbon processes and all nitrogen processes are computed yearly. The schematic illustrates the role of meteorological data inputs, derived data such as *leaf-area index* (LAI), and individual subprocess models such as Snow (for snowmelt), Transpiration, and Photosynthesis.

Forest-BGC, as depicted in Figure 2, is point-based: it describes processes at a single point. We expand the physical scale of the model by combining point-based computations for a collection of contiguous points (or more precisely, for contiguous pixels of finite size), and by adding new model components that describe interactions between points. Such an increase in scale dramatically increases the model's computational demands. For example, increasing the study area from a number of isolated points to a two-dimensional square area increases the number of pixels to n^2, where n is the length of a side of the area, in pixels. More importantly, in the most general case where any point can interact with any other point, the worst-case complexity for computing point-point interactions is very roughly of order n^4, or $O(n^4)$ in the standard notation of complexity analysis. Similarly, increasing the study area to three dimensions results in a pixel count on the order of n^3 and worst-case complexity of roughly $O(n^6)$ for computing possible interactions. In terrestrial systems most forms of interaction are limited, so worst-case complexity is rarely realized. However, given the very large size of n in even modest regional-scale studies, an important aspect of model design is developing point-point interaction models with much smaller computational costs.

To illustrate, soil hydrology in the Forest-BGC point model is based on a simple "bucket" analogy, where the amount of available water is determined by water coming to that point as input, constrained by the size of a bucket representing local water-holding capacity. This approach, however, does not account for horizontal redistribution of soil water, which often dominates soil hydrology, especially in mountainous terrain. For a more realistic model we need to add a component that describes point-point interaction for both above-ground and below-ground water flow. A maximum-complexity flow model might assume that excess water could flow from one model cell to any other cell in the three-dimensional space, but a more practical model would simply embody the principle that excess water flows from higher areas to lower areas. A module called Topmodel[3,5] adds this practical treatment of lateral water flow, based on off-line analysis of topography (for surface flow) and soil properties (for subsurface flow). Topmodel uses basic topographic data, such as a digital map of elevations, to partition the landscape into static areas that represent collections of individual pixels, thus reducing the set of point-point interactions to a much smaller set of area-area interactions. Similarly, for below-ground flow Topmodel uses a *topography-soils index* of similarity to stratify zones of differing soil-water dynamics. Any two points with the same similarity index are then considered hydrologically identical.

To compute the area-area interactions of water flow, Topmodel uses the "buckets" signifying holding capacity, the flow direction (downhill along the topographical gradients),

measurements of the landscape unit or watershed area that collects above-ground flow, and the predicted volume of that flow. Simply put, Topmodel moves excess water from one watershed area to another, rather than trying to move it from pixel to pixel. Topmodel enhances Rhessys in two important ways: it allows a better estimate of both the timing and quantity of stream discharge (the volume of water flowing in a stream or river), and it provides a better representation of soil moisture patterns across the landscape, which has a strong influence on rates of evapotranspiration and photosynthesis. In computational terms, the key aspect of Topmodel is that it uses abstractions for landscape and soil values that dramatically reduce the complexity of computing point-point interactions in the three-dimensional model space. We compute the abstractions once, off-line, for a particular study area, and can then use them unchanged in any number of simulation experiments.

Expanding a modeling system can give it great potential, but make actual use difficult.

The overall simulations Rhessys does are driven not only by process-level models of vegetation and soils, but also by accurate meteorological data at daily time resolution or greater. Reliance on meteorological process models brings two new types of complexity into the modeling system. Unlike models for vegetation, soils, and topography, which are based on relatively static input data, weather processes are based on air temperatures, barometric pressures, incoming solar radiation, and precipitation that change at least daily. Also, no matter what scale of modeling we are doing, weather data are available only for a few measurement sites, not for all points in the study grid. Together, these data characteristics impose severe information system constraints. It is essential to use some form of standard meteorological data source. Although many government agencies collect weather data for specific purposes, only the Environmental Data Service of the National Oceanic and Atmospheric Administration reports consistent and widely available data. In practice, even these standard data have limitations. For example, the standard daily data collected by the US National Weather Service and cooperating stations are limited to precipitation and maximum/minimum temperature. Standard data are not available for many mountain sites, as most regular weather stations are in populated areas, which tend not to be in the mountains. Finally, meteorological data specific to ecological processes, such as the length of time each day that solar radiation is above the threshold needed for photosynthesis, are not available from any routine source.

Rhessys uses the meteorological process driver MT-Clim, which can build a complete daily meteorological map for a mountainous or otherwise topographically diverse study region from a limited set of direct National Weather Service measurements. The centerpiece of MT-Clim is a submodel that calculates daily potential incoming solar radiation for any elevation, slope, directional aspect of slope, and latitude. As illustrated in Figure 3, MT-Clim uses the observed difference between maximum and minimum temperatures during a day to calculate an atmospheric transmissivity (which incorporates cloud cover), eliminating the need to use the inaccurate visual cloud-cover estimates provided by the NWS. We then apply elevation and aspect-based corrections to estimate key values such as surface temperature, dew points, and precipitation. Validation studies of this model on three north-south mountain slope pairs in the state of Montana indicate that these relatively simple climatological calculations produce data accurate enough for simulation of seasonal evapotranspiration and photosynthesis in forests.[6] In information system terms, a key aspect of MT-Clim is that it lets us use an electronically available, standard input source for our highest-volume data input stream. MT-Clim also provides Rhessys with flexibility of scale, and it lessens model dependence on hard-to-obtain, expensive, and highly variable field measurements.

A particularly important property of modeling systems, and one we have set as a goal in the continuing development of Rhessys, is flexible scaling. Ideally, a researcher should be able to expand or contract modeling-system scale simply by modifying system parameters and selecting appropriate processing components. Flexible scaling increases the range of applications for a model, allowing the same system to be used to study a particular stand of trees, a watershed, an entire ecosystem, or a multi-ecosystem region. Of course, researchers must plan for computational costs that increase proportionally with scale, and they must create point-point interaction models that are realistic, scalable, and as simple as possible to compute.

A more subtle need is for a system that can easily adapt to input data and computational processes whose basic form varies with model scale. For instance, most researchers have calculated the flux of energy and mass from terrestrial ecosystems at one of two dramatically different spatial scales. At one extreme, global-scale computer models for climate[7] or the global carbon cycle[8] treat the earth's entire land surface as a single unit or as a collection of discrete units of about 1 degree latitude by 1 degree longitude. At the other extreme lie plant-level and stand-level process models of ecosystems.[9] These micromodels can rely on field measurements to determine leaf density or mass, the primary factor driving processes such as photosynthesis and evapotranspiration. In a regional study, it isn't feasible to collect enough detailed field measurements for a large area with highly varied vegetation. A more abstract measure of plant density, such as leaf-area index, might work, but the researcher must find or compute LAI values at a resolution and accuracy appropriate to the regional scale. Thus, even though two models at different scales may try to balance the same element budgets, the way in which processes are modeled and the inputs required may differ dramatically.

To put the effects of variable scaling in more general terms, consider a modeling system made up of three basic process models, P_A, P_B, and P_C. Assume that for a small study scale α_1, implementation modules M_A, M_B, and M_C perform the process-level computations using derived data inputs D_A, D_B, and D_C. Further assume that we produce D_A, D_B, and D_C off-line by using input preprocessing steps I on raw data inputs R_A, R_B, and R_C. That is, $I_A[R_A] \rightarrow D_A$, $I_B[R_B] \rightarrow D_B$, and $I_C[R_C] \rightarrow D_C$. When we increase the study area to a larger scale α_2, a chain of revisions is needed. First, some or all of the process models must be adapted to the new scale, giving rise, for example, to new component versions $M_{\mathcal{A}}$, $M_{\mathcal{B}}$, and $M_{\mathcal{C}}$. Second, inputs must be scaled up. The raw data required to produce the derived data sets may or may not be available at the larger scale. If they are available, we can develop and apply a new version of the preprocessing step; a new $I_{\mathcal{A}}[R_{A1}, ..., R_{An}] \rightarrow D_{\mathcal{A}}$, for example, is all we need. If appropriate raw data are not directly available, we may have to develop an entirely new subprocess pipeline to create and use new derived data sets. Such a pipeline would require the following new components: logical process model $P_{\mathcal{B}}$, implementation module $M_{\mathcal{B}}[D_{\mathcal{B}}]$, raw data sets $R_{\mathcal{B}1}, ..., R_{\mathcal{B}n}$, and input process $I_{\mathcal{B}}[R_{\mathcal{B}1}, ..., R_{\mathcal{B}n}] \rightarrow D_{\mathcal{B}}$. This proliferation of components adds much to the complexity of the information system, and will soon confuse users.

The need to use different source inputs and model components for different scales makes clear the need for a coherent information system. As the number of types of source inputs and versions of processing components grows, the number of ways they can be combined into a complete modeling system grows dramatically. The time spent determining which raw or derived data sets should be used in a particular simulation also increases. Researchers

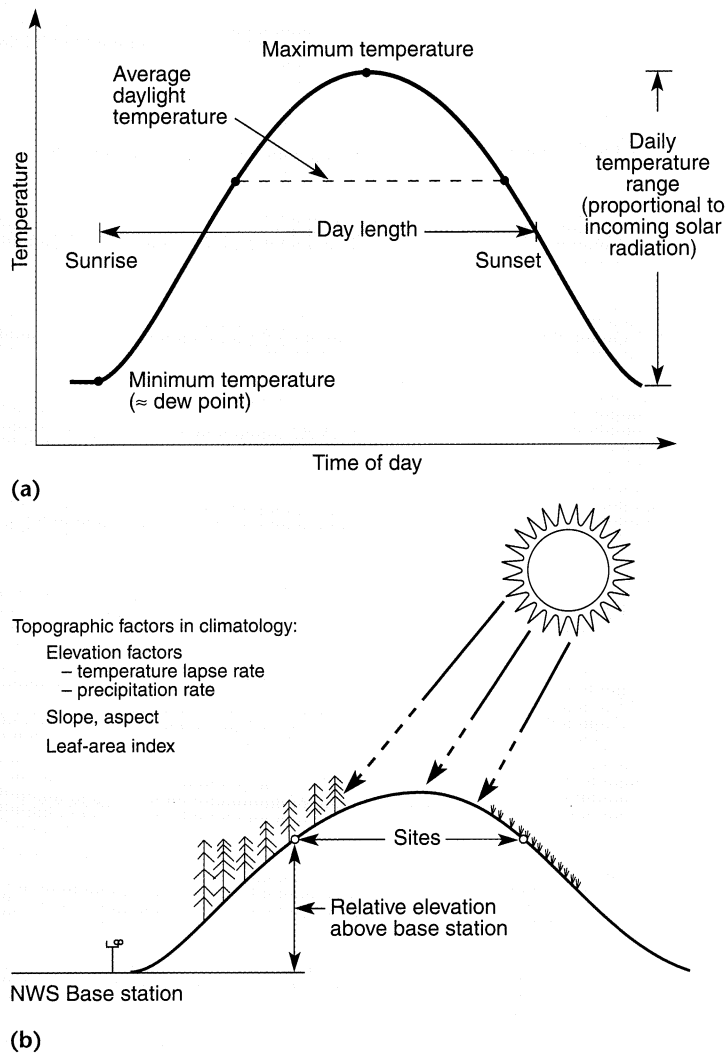


Figure 3. The MT-Clim submodel uses simple standard weather data to calculate the complex daily microclimate information that Rhessys needs. In (a), daily temperature range is used to measure total solar radiation, and daily minimum temperature is used to estimate humidity. In (b), elevation effects, such as lower temperature and higher snowfall as you go up a mountain, are used along with other topographic factors to correct base-station data when modeling the climate of each specific site.

commonly observe that although expanding a modeling system gives it great potential to do lots of things, it becomes more and more difficult to get the system to do any particular thing. The capabilities are there but the system is no longer user-friendly. Good information system design should allow the modeler to easily identify the submodel components and data sets that are available and appropriate to use at the various scales. While cheaper and faster computers can help lessen the impact of scale increases on computation cost, only an improved information system can help us deal with the proliferating data and components.

Rhessys: The computational pipeline

To actually compute a Rhessys simulation, we start with off-line input preprocessing. This transforms various low-level input data sets into

Computing Leaf and Soils Data

A critical aspect of our ability to scale Rhessys to a variety of applications is the identification of accurate and accessible characterizations of both vegetation and soils attributes for the land units in our study area. In general, we abstract the properties of particular biome types (for example, forest, grassland, or cropland) with the concept of *leaf-area index*, which roughly counts the layers of leaves through which moisture or sunlight must pass before reaching the land surface. Forest-BGC uses LAI as a principal vegetation characteristic, central to calculating interception of rainfall and radiation, transpiration, and photosynthesis. The science of remote sensing has proved invaluable in determining LAI. We can estimate LAI values efficiently for large or small areas, at a variety of resolution scales, by automatic analysis of satellite imagery.

For example, two of us (Ramakrishna Nemani and Steven Running) observed a strong curvilinear relation between ground-measured LAI and remotely sensed values of the normalized-difference vegetation index for 53 conifer stands in Montana.[1] The NDVI is a calculated relationship between the amounts of radiation reflected by plant life in two segments of the spectrum: near-infrared and visible red. In this case the data were from the NOAA's Advanced Very High Resolution Radiometer satellite-based instruments. Another research team, which included one of us (Running), got similar results using the Thematic Mapper sensor of the Landsat satellite, which operates at much higher spatial resolution than the AVHRR. Here the ratio of near-infrared to red radiation reflected by vegetation and received by certain channels of the TM sensor showed a 0.91 correlation with the measured LAI of coniferous forests.[2] Other investigators have also found that LAI correlates well with the near-infrared/red reflectances of vegetation.[3] The ability to derive this abstraction of the vegetation component for large areas automatically, instead of having to rely on laboriously collected field data, greatly increases the scalability of the Rhessys model. More details on the use of remote sensing to derive LAI can be found in the literature.[3,4]

Another class of critical ecosystem inputs that have been traditionally difficult to obtain is a characterization of surface soil, in specific terms of its moisture-holding capacity. Soil data sets are generally available only at very large scales and do not contain enough information to let us compute moisture-holding capacity. We have thus been forced to estimate the required properties by other techniques. At present we approximate by adjusting the available coarse-resolution soil maps in two ways. First, we use remotely sensed observations to identify talus and rock outcrop areas which we associate with "shallow" soil depth. We assign "low" water-holding capacity and "high" water transmissivity values to these areas. Next, we use topographic data to estimate soil depth, and corresponding water-holding and transmissivity values, ranging from the "shallow" soil at the at ridgetops to "deep" soil in valley bottoms. We are now developing a more ambitious soil property inferencing system, which uses additional remotely sensed data, geological maps, and topographic position, combined with rules for the soil-climate-topography-vegetation associations.

The last stage of preparation for these inputs is to geolocate the remote sensing and soils data to the topography. While deriving LAI from remotely sensed data, topographic data is used to correct the vegetation indices. Topographic information is also used to adjust the coarse-resolution soils information as described earlier. Finally, the soil and topographic information is combined to produce a *topography-soils index*, $\ln(\alpha T_e)/T_i \tan\beta$. While α (the drainage area that is upslope of each pixel) and β (the slope of a given pixel) are derived from digital terrain analysis, T_e (mean transmissivity of soil for the landscape unit) and T_i (transmissivity of a given pixel) are derived from the soil maps.

References

1. R. Nemani and S. Running, "Testing a Theoretical Climate-Soil-Leaf Area Hydrologic Equilibrium of Forests Using Satellite Data and Ecosystem Simulation," *Agricultural and Forest Meteorology*, Vol. 44, 1989, pp. 245-260.
2. D. Peterson et al., "Relationship of Thematic Mapper Simulator Data to Leaf Area Index of Temperate Coniferous Forests," *Remote Sensing of Environment*, Vol. 22, 1987, pp. 323-341.
3. R.B. Myneni, G. Asrar, and S.A.W. Gerstl, "Radiative Transfer in Three Dimensional Canopies," *Transport Theory in Statistical Physics*, Vol. 19, 1990, pp. 1-54.
4. G. Asrar, R.B. Myneni, and B.J. Choudhury, "Spatial Heterogeneity in Vegetation Canopies and Remote Sensing of Absorbed Photosynthetically Active Radiation: A Modeling Study," *Remote Sensing of Environment*, Vol. 41, 1992, pp. 85-101.

the higher-level abstractions required by the Rhessys process models. One such pipeline computes the leaf-area index abstraction of actual vegetation by analyzing remotely sensed data. This use of electronically available data to replace sparse, expensive-to-obtain field vegetation measurements is critical to scaling the simulation to regional and larger levels (see sidebar on p. 38).

As we mentioned earlier while discussing Topmodel, in a second important pipeline we create the high-level network of abstract land units, replacing the original complete grid of regular pixels. The dominant physical processes of energy and mass exchange, transpiration, and photosynthesis are characterized by strong nonlinearities, so simple areal averaging cannot represent a complex landscape.[2] This preprocessing first produces the landscape template by grouping pixels into larger discrete areas, and then organizes data for topography, soils, and remotely sensed attributes according to the newly defined landscape units. This is done using principles from GIS, or geographic information systems, a computerized mapping technology in which each defined spatial region can be linked to multiple attributes stored in a relational database.

The primary objectives for landscape aggregation (grouping pixels into areas of like character) are

(1) minimizing within-unit variance so that each landscape unit can be assumed to be homogeneous, and
(2) maximizing between-unit variance of parameter fields so as to accentuate the difference in landscape ecological activity.

The shape of a hill or mountain can be approximated by a set of flat facets, much like a complex cut gemstone. Each facet, having a slope, elevation, and directional aspect that differ from surrounding facets, could be expected to have slightly different ecological characteristics. In our study sites in western Montana, the strong influence of topographically controlled microclimate on surface ecological processes results in a geographic pattern of forest cover that closely follows the pattern of hillslope facets. Consequently, we use abstract topographic partitions to define the land surface as a connected collection of drainage areas, and then use those partitions as templates to organize all other input variables. We are able to automatically construct the appropriate land unit definitions at variable scales through automatic analysis of digital terrain data,[10] thus supporting our overall modeling goal of flexible scaling. The target scale resolution is determined by the value selected for the minimum size of a "drainage area." The larger the minimum-size drainage area is, the smaller the number of drainage areas will be. Topographic analysis then allows us to organize the collection of drainage areas into a partitioned drainage network for the entire study area.

Once preprocessing is complete and final vegetation and soils inputs are referenced to the landscape, the Rhessys simulation run begins. Processing generally begins at ridgelines and follows topological flow into valleys. The core process model, Forest-BGC, is initialized at the beginning of each run with generic physiological parameters for conifer forests. At the end of each simulated day, a new mean soil-water deficit is calculated for the landscape unit. A typical simulation continues for an entire simulated year, passing through the complete water recharge/release cycle common to many mountainous areas.

Rhessys produces several kinds of output products. Variables like snowpack, soil water, stream discharge, transpiration, evaporation, and photosynthesis can be calculated for each landscape unit and for each TSI interval, for selected days or a whole year. In addition, Rhessys can generate images showing spatial variation across the landscape in soil moisture, snowpack, evapotranspiration, and photosynthesis for any given day. Figures 4 and 5 show recent Rhessys simulations of the Seeley-Swan valley in northwestern Montana, an area of about 100 by 50 kilometers. Figure 4 shows landscape units false-colored according to their computed LAI values, then overlaid on a digital elevation map to produce a 3D image. This area has two major mountain ranges, the Mission mountains on the left (west) and the Swan mountains on the right (east). The mountains produce strong precipitation and temperature gradients that have a pronounced climatic influence on vegetation (mainly coniferous forests) on both the mountain slopes and valley floor. Precipitation varies from 40 cm/yr at the southwest corner to about 200 cm/year at the northeast corner of the area shown. Extensive clearcutting of forests in this area over the last few decades has had a major effect on the ecosystem. Clearcutting produces low LAI values, and is easily recognizable in the figure.

Figure 5 illustrates the flexible scaling of Rhessys, in typical outputs. The images in this figure are all based on our abstract land unit

partitioning, with color added to indicate annual evapotranspiration rates predicted by Rhessys. In the first image the study area was partitioned into five sections, representing large drainage areas. In the second image we defined 170 partitions for the same area. Though the absolute magnitudes of evapotranspiration computed at each partition level are similar (mostly 50–70 cm/yr), there is considerably more spatial detail. The flexibility of scale is illustrated even further with similar calculations performed for Soup Creek, a 13-square-kilometer drainage within the study area. Soup Creek cannot be distinguished in the five-partition image. Even in the 170-partition image it is represented by only two partitions. However, when resolution is increased to the point where Soup Creek alone has 22 partitions, the attributes of individual portions of that watershed become obvious. We can get a valuable range of scientific results by varying the scale or resolution and then executing the same basic simulation. The range of images shown in Figure 5 illustrates the changing importance of factors controlling evapotranspiration rates as we telescope from an entire region down to a single watershed. At regional scale (five partitions), climatic factors, precipitation, and temperature exert strong influence on ET rates. At the other extreme (in a single watershed), microclimate, along with lateral redistribution of soil water owing to fine differences in topography, strongly influences ET patterns. The ability to correctly represent the controlling factors at each scale has important implications. For example, mesoscale climate modelers may find the spatial patterns of ET computed at 170 partitions appropriate for comparison with ET estimates from their own models. On the other hand, a forest manager interested in increasing water yield of a certain drainage area or watershed would like to see spatial patterns of ET at watershed scale. More details on the landscape partitioning, and on "ground truthing" of Rhessys model results, appear elsewhere.[3]

In developing Rhessys, we have had considerable success in finding reasonable solutions to the input substitution problems that arise as study areas get larger. Our larger, faster workstations can now complete bigger and longer simulations in a fraction of the time originally required. Unfortunately, because of the ad hoc nature of our current information system, it also takes us much longer to prepare our simulations for execution. Simply finding the data, preprocessing them appropriately, and assembling the set of appropriate computation elements now takes somewhere between 75 and 95 percent of

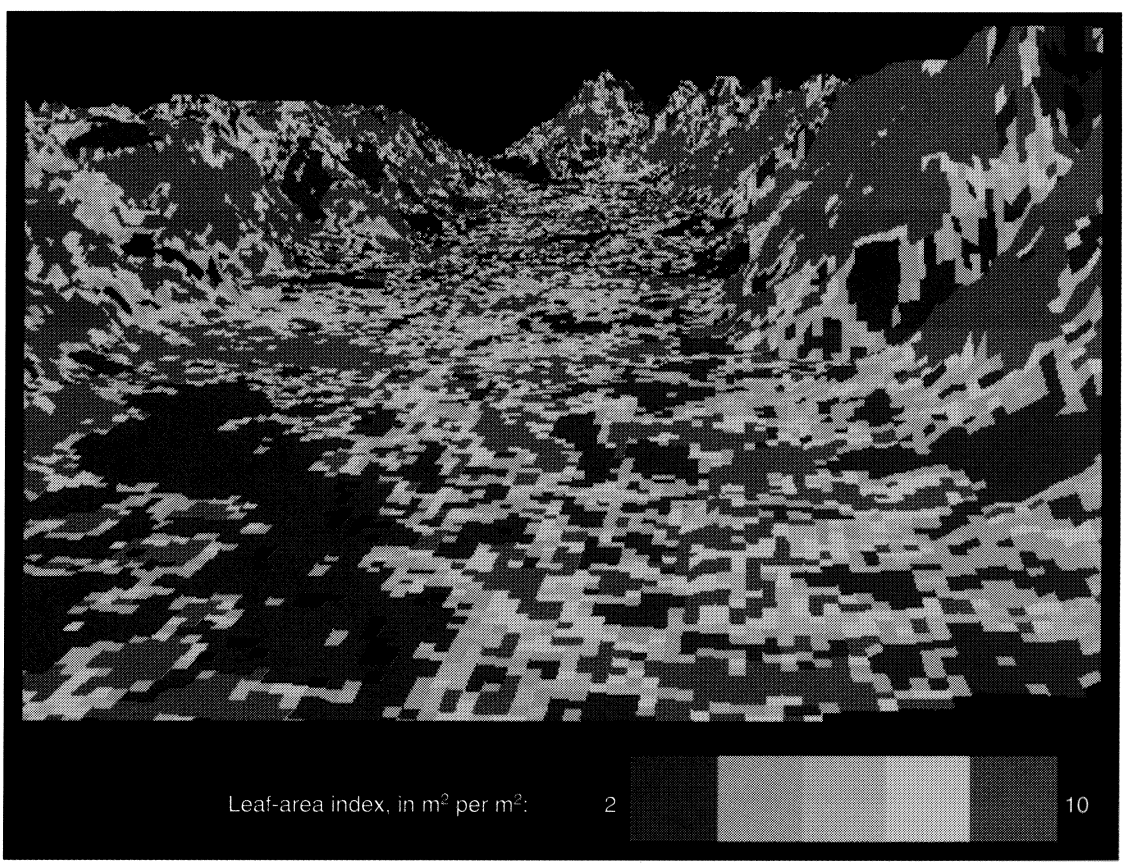


Figure 4. Rhessys can use satellite data to compute a *leaf-area index*—a density measure of leaves per unit area—and overlay this on a 3D map. This simple simulation of the Seeley-Swan valley in Montana shows the effects of precipitation and forest clearcutting on vegetation density. Leaf-area index, in turn, is one of several data sets used by Rhessys to simulate more complex ecological processes.

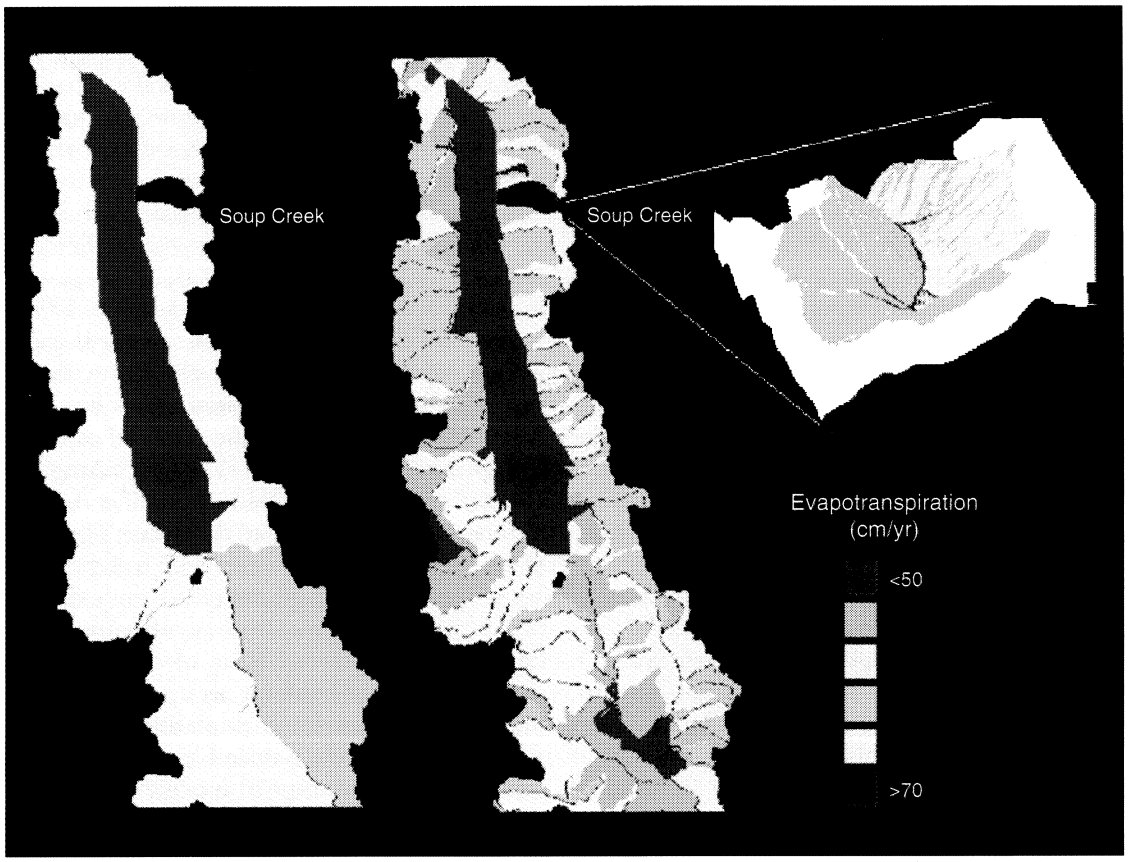


Figure 5. In a complex simulation, Rhessys models evapotranspiration of water into the air from forests in the Seeley-Swan valley. Set at a gross scale (*left*), the model creates only a few partitions, shown by colors representing different transpiration rates. At a medium scale (*middle*), many more landscape partitions show increasing detail (dark dotted lines separate, along ridgelines, landscape partitions with the same rate). A computational enlargement of one small watershed (*right*) shows even more detail at a very fine scale.

the modeler's time. The origin of this bottleneck is the problem described earlier: how does a modeler assemble all the "right" data sets and computational components from the many options available?

An ecosystem information system evolves

As noted earlier, the present Rhessys is increasingly burdened by the baggage of its own successes—a myriad of model components, raw and preprocessed data sets, study results, specialized models, and so on. A critical next step in the evolution of Rhessys is to reorganize this wealth of scientific material into a more accessible ecosystem information system. We call this system EIS, which, though it takes the same acronym, should not be confused with the environmental impact statements required by federal law before construction projects. The adaptation of Rhessys to our EIS is now well underway, based on object-oriented design and implementation principles. The object-oriented paradigm is particularly apt in organizing Rhessys components. Because virtually all components share a large number of common spatial attributes (that is, they represent abstractions drawn from similar 2D and 3D spaces), they fall quite naturally into a well-defined hierarchy of classes, instances, and methods. The goal of system software for EIS support is to build a robust distributed system that provides access to our object-oriented view of the information resource. EIS functions will allow a researcher interested in simply using the information resource to browse a class-hierarchy, select processing components, specify required parameters, verify the consistency of parameter specification, and manage the execution of an operation (which includes accessing data and creating data objects). For researchers interested in extending the information resource, EIS will support the addition of class and instance definitions, and the association of methods with operation definitions. Finally, EIS will also assist in the development and use of efficient data management techniques.

As illustrated in Figure 6, our EIS design is based on a distributed object-requestor/object-broker paradigm, with four primary subsystems. An object system manager (OSM) is the overall system manager, responsible for system startup, orderly termination, and object persistence (that is, logging to persistent storage). Instances of graphical user interface (GUI), object request broker (ORB), and object database manager (ODBM) processes execute on each participat-

ing workstation, screening user-initiated object requests, resolving requests locally whenever possible, and "broadcasting" other requests for global resolution when necessary. In our initial version, the ODBM provides only local support for object storage and retrieval. Because of the large number and size of objects already in the system (objects up to 400 megabytes are not uncommon), developing a more ambitious ODBM with global responsibility for physical object placement is a longer-term project. Our first version performs only an "object registration" function. That is, it permits objects to be registered as instances in the class hierarchy and permits users to browse the hierarchy and extract basic object attributes, but does not support automatic storage or retrieval of complete objects.

Translating the ORB paradigm into effective practice involves several important decisions, including what strategy to use in determining where objects and methods are stored, how to facilitate ORB decisions regarding local versus global resolution of object requests, and what communication mechanisms to use in the distribution protocol. Unlike commercial object-oriented information systems designed to handle many requests for relatively small objects and short-duration object operations, we expect the workload in our research environment to be dominated by class-hierarchy browsing, along with relatively infrequent but long-duration operations on relatively large objects. Our plan focuses on putting the object support system in place quickly to establish the object-oriented paradigm among Rhessys users. To that end our first version uses a simple, high-level mechanism to implement the distribution protocol. Once we have the opportunity to analyze the workload more extensively, we will build more efficient and robust distribution protocols into future versions. The sidebar on page 43 further discusses some practical considerations we are taking into account in the ongoing design of EIS.

We have come to realize that we need visualization capabilities to effectively deal with the vast amounts of data our simulations can produce. We are building in the required capabilities as part of our system redesign, taking advantage of existing visualization software that characterizes complex spatial objects. The visualization package we are currently using, IBM's Visualization Data Explorer (DX), embodies an object-oriented approach to visualization based on a general characterization of n-dimensional spaces. The Data Explorer also provides a simple textual syntax for formulating object descriptions. EIS uses comparable object-description syntax and conceptual models for defining its own spatial concepts as ecosystem classes and objects. By starting with the IBM package's syntax and its high-level classification of spatial concepts, we avoided redundant effort in designing object-description syntax. With this base we have provided direct links to the visualization capabilities of the Data Explorer early in EIS development. We are also currently writing a number of format conversion programs to

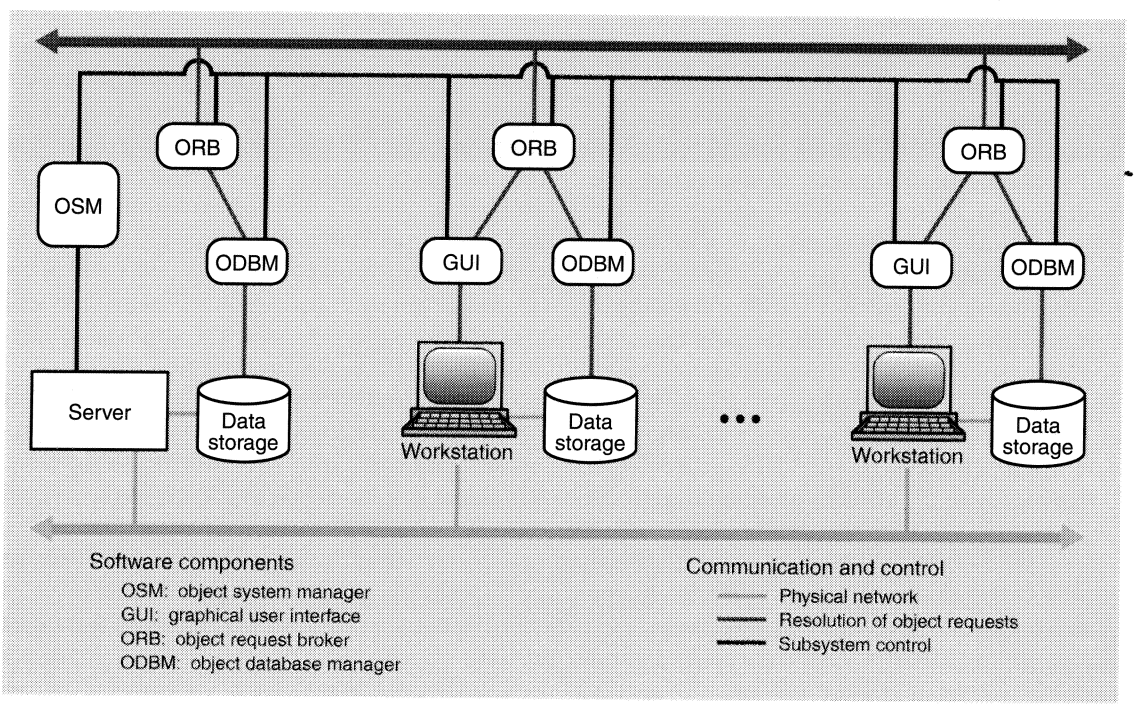


Figure 6. Our Ecosystem Information System, shown in schematic, takes an object-oriented approach to helping users make sense of the proliferating data sets and models in Rhessys. As Rhessys has been scaled up and made more powerful in an ad hoc way, increasing user confusion has made better organization essential.

User Expectations Guide the Design of EIS

It has been almost four years since we first formulated our vision of a distributed "Ecosystem Information System" that would use Internet facilities to allow data and models to be shared among ecosystem modelers in labs scattered around the world. Four years ago few of our intended users had much experience with network-based "information sharing" tools, and thus had few expectations of how such tools would work and what graphical user interface they might provide. Similarly, as developers we could assume that few tools were available to support the distributed system programming effort, other than basic network programming facilities such as sockets.

User expectations have by now changed dramatically. Information-sharing tools have become widespread, and hypertext-based graphical interfaces, such as Mosaic, have become the de facto standard for how such systems should look and operate. We find that our intended users now assume that a distributed information-sharing system like EIS will look and feel like Mosaic, and they react negatively if it does not. Furthermore, if possible they want to enter, or "launch," EIS from Mosaic in a more or less transparent manner. This emergence of Mosaic as a de facto standard has changed the GUI design for EIS, and in fact has made our task simpler by providing a GUI model that we can emulate.

At the implementation level, the situation is much less clear. Over the last four years a number of high-level mechanisms have been proposed to better support distributed system programming. These include various forms of remote procedure call, distributed object environments, and object request brokers. Unfortunately, most of these are commercial products, and no single mechanism has emerged as dominant across multiple platforms. A requirement for EIS is that it must be "freely" available and supported across a wide range of vendor platforms. Thus, it isn't feasible for us to build the EIS system using facilities that require users to buy additional run-time support libraries, or that are only supported on a few platform types. Therefore, while we have done some prototyping with some of the higher-level distributed programming facilities, our initial version of EIS is based only on the lowest common denominator of net services: sockets and socket-programming services.

translate data sets in various standard formats into the object syntax used by DX and EIS.

Our long-term goal for Rhessys is to use it in increasingly complex large-scale studies such as the global 1×1–degree mapping of primary ecosystem production shown in Figure 7. Global studies demand dramatically higher computational power. They also create a pressing need for good data management systems such as EIS. Over the years we have moved from running relatively small models using small data sets on PCs, to increasingly larger models and data sets on powerful workstations. Data-set preprocessing and data management requirements have also expanded greatly. For example, the Rhessys simulation in Figure 7 was modeled by about 14,000 pixels. It took about

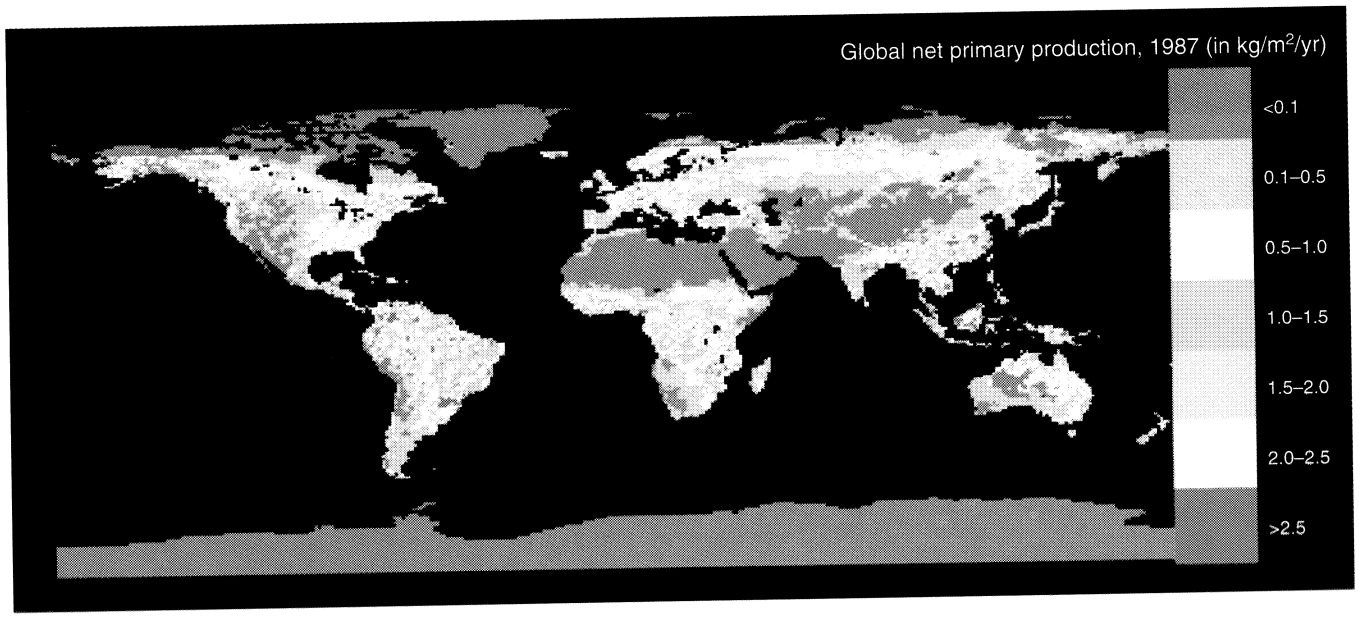


Figure 7. Rhessys principles can be scaled up to global-scale modeling efforts.

4.5 hours to compute on an IBM RS/6000 model 350. Before actually running the global study, we had to extrapolate daily climate data from 6,000 weather stations to the global grid, preprocess global satellite data into a monthly-basis form, and adapt the lower-level process model Forest-BGC to the variety of vegetation types found worldwide.

A Rhessys simulation now in development models the entire Columbia River Basin in the northwestern United States. Because of the fineness of scale this model is computationally much larger even than the global simulation; it uses about 600,000 pixels. Actual Rhessys simulation time is expected to take 50 to 60 hours on our RS/6000-350 workstation, while data-set preprocessing will occupy a number of people and workstations for approximately three months.

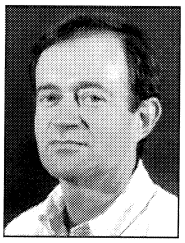
Rigorous estimates of net primary production at a global scale are needed to assess the role of vegetation in such key contemporary issues as climate change and global habitability. Comparable estimates at large regional scales are important in assessing more immediate socioeconomic issues such as land use and development practices. We hope that by continuing to refine Rhessys modeling components and by enhancing the information system that comprises the "Rhessys environment," we will meet these new modeling challenges well into the next decade. ◆

Acknowledgments

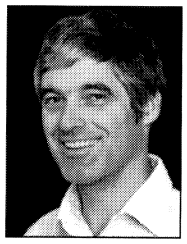
Funding for this research has been provided by grants NASA/NAGW-952, NSF/BSR-8817965, and NSF/BIR-9301760. We acknowledge the contributions of Lars Pierce, Larry Band, Richard Lammers, and other members of the Numerical Terradynamics Simulation Group, as well as Roly Redmond, Ron Righter, Jack Cummings, and other members of the EIS design team. The comments and suggestions of the anonymous referees were also very helpful.

References

1. S. Running et al., "Mapping Regional Forest Evapotranspiration and Photosynthesis by Coupling Satellite Data with Ecosystem Simulation," *Ecology*, Vol. 70, 1989, pp. 1,090-1,101.
2. L. Band et al., "Forest Ecosystem Processes at the Watershed Scale: Basis for Distributed Simulation," *Ecological Modelling*, Vol. 56, 1991, pp. 171-196.
3. L. Band et al., "Forest Ecosystem Processes at the Watershed Scale: Incorporating Hillslope Hydrology," *Agricultural and Forest Meteorology*, Vol. 63, 1993, pp. 93-126.
4. S. Running and J. Coughlan, "A General Model of Forest Balance, Canopy Gas Exchange and Primary Production Processes," *Ecological Modelling*, Vol. 42, 1988, pp. 125-154.
5. K. Beven and M. Kirkby, "A Physically Based, Variable Contributing Area Model of Basin Hydrology," *Hydrologic Science Bull.*, Vol. 24, No. 1, 1979, pp. 43-69.
6. S. Running, R. Nemani, and R. Hungerford, "Extrapolation of Synoptic Meteorological Data in Mountainous Terrain, and its Use for Simulating Forest Evapotranspiration and Photo-synthesis," *Canadian J. Forest Research*, Vol. 17, 1987, pp. 472-483.
7. S. Manabe and R. Wetherald, "On the Distribution of Climate Change Resulting from an Increase in CO_2 Content of the Atmosphere," *J. Atmospheric Science*, Vol. 37, 1980, pp. 99-118.
8. W. Emmanuel et al., "Modeling Terrestrial Ecosystems in the Global Carbon Cycle with Shifts in Carbon Storage Capacity by Land-Use Change," *Ecology*, Vol. 65, 1984, pp. 970-983.
9. W. Parton et al., "Analysis of Factors Controlling Soil Organic Matter Levels in Great Plains Grasslands," *Soil Science Soc. of America J.*, Vol. 51, 1987, pp. 1,173-1,179.
10. L. Band, "Spatial Aggregation of Complex Terrain," *Geographical Analysis*, Vol. 21, 1989, pp. 279-293.



Ray Ford is an associate professor of computer science at the University of Montana. Earlier he was on the faculty of the University of Iowa and the University of Kansas. He has a BS and MS from the University of Missouri, Rolla, and a PhD in computer science from the University of Pittsburgh. His research interests include distributed systems and applications, concurrency control, and scientific visualization. He is currently combining these interests in approaching the sort of ecosystem modeling, analysis, and support projects described in this article, with support from NSF, DOE, and a collection of private companies. Ford is a member of IEEE and ACM.



Steven Running trained as a terrestrial ecologist, earning his BS and MS at Oregon State University and his PhD at Colorado State. He has been at the University of Montana since 1979, where he is now a professor of forestry. His primary research interest is the development of global and regional ecology models by integrating remote sensing with climatology and terrestrial ecology using computer simulation. Running is a member of the Scientific Steering Committee for the Biospheric Aspects of the Hydrologic Cycle project of the International Geosphere-Biosphere Program. He is also a team member for the NASA Earth Observing System, Moderate Resolution Imaging Spectroradiometer (EOS/MODIS).

Ramakrishna Nemani is a research assistant professor at the University of Montana. Following agronomy and meteorology studies in India, where he earned a BS from A.P. Agricultural University in Hyderabad and an MS from Punjab Agricultural University in Ludhiana, Nemani obtained a PhD in forest ecology at the University of Montana in 1987. His current interests include extraction of biophysical information about land surfaces (leaf-area index, surface wetness, and land cover) from remote sensing observations, and ecosystem modeling at continental to global scales. Nemani is an associate team member for the NASA EOS/MODIS project and is a member of the American Geophysical Union.

Ford can be reached at the Department of Computer Science, University of Montana, Missoula, MT 59812, e-mail ford@wilfred.cs.umt.edu. Running and Nemani are at the School of Forestry, University of Montana, Missoula, MT 59812; swr@hps1.ntsg.umt.edu and nemani@hps1.ntsg.umt.edu.