# Conceptual user interface for the land management system

James Westervelt and Jeffery Holland

## ABSTRACT

This paper explores the conceptual user interface requirements of the Land Management System (LMS), a next-generation system designed to support the development of location-specific landscape/watershed management oriented simulation models. Currently available landscape/watershed models tend to be discipline-specific, focusing only on hydrology, ecology, social, economic or agronomic aspects of the landscape's subsystems. Feedback loops among the different subsystems tend be ignored, and this can result in long-term predictions that may not be useful. LMS will provide landscape and watershed managers with sets of software modules that can be linked together to represent and simulate unique local conditions. A design challenge of LMS is to develop a user interface that makes it possible for a watershed/landscape manager to develop and use multidisciplinary spatially explicit landscape simulation models that retain the scientific rigour of current scientist-oriented simulation models. This paper outlines a solution in response to that challenge.

**Key words** | landscape, modelling, multidisciplinary, simulation, watershed

**James Westervelt** (corresponding author)
Construction Engineering Research Laboratory,
US Army Engineer Research and Development Center,
US Army Corps of Engineers,
2902 Newmark Drive,
P.O. Box 9005,
Champaign,
IL 61826-9005,
USA
Tel.: +1 217 352 6511
E-mail: *j-westervelt@cecer.army.mil*

**Jeffery Holland**
Information Technology Laboratory,
US Army Engineer Research and Development Center,
US Army Corps of Engineers,
3909 Halls Ferry Road,
Vicksburg,
MS 39180-6199,
USA
Tel.: +1 601 634 2644
Fax: +1 601 634 3193
E-mail: *hollanj@mail.wes.army.mil*

## HISTORICAL BACKGROUND

The management of landscapes and watersheds normally involves consideration of multiple goals involving hydrologic, ecologic, social, economic and agronomic objectives. Managers deal with (1) ideas about what needs to be fixed, changed, or maintained in the natural resource being managed, and (2) ideas about actions proposed to address those objectives. These ideas are displayed in Figure 1 as two lists. The rightmost identifies desired outcomes that a manager might be addressing and is matched in the leftmost list with a set of proposed actions. These outcomes and actions, in the most general sense, vary over time and location. The question mark in the middle of the figure identifies the primary modelling goal: what will the impact of a set of actions be with respect to desired outcomes as a function of time and location? Models of the system can help evaluate the risks and consequences associated with the proposed actions with respect to the desired outcomes. Today, informal conceptual models developed in the minds of managers, residents, and stakeholders provide the best comprehensive multidisciplinary understandings of the entire system. Formal scientific models capture the detailed understandings of the processes associated with respect to small parts of the whole system. A primary goal in regional planning today is to inexpensively develop formal models of the entire system so that important feedback loops among the different components of the system are captured in a manner that allows the model to accurately indicate the implications of alternative management options. Such a system must fully embrace the goal of allowing users to identify alternative management suggestions (left-hand list in Figure 1) and consequences of interest (right-hand list in Figure 1).

Computer simulation modelling has been used successfully in support of landscape/watershed management. However serious challenges severely limit the use and

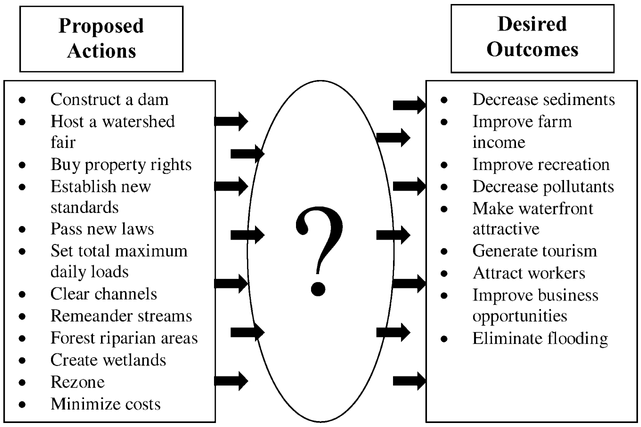


**Figure 1** | A management conceptual view.

impact of simulation models. A short review of this history will help develop an understanding of the challenges involved in the design of a conceptual user interface for a comprehensive multidisciplinary modelling system. A number of different disciplines are involved in the management of natural and human resources at the watershed and landscape scales. These include hydrologic engineering, urban and regional planning, regional economics, landscape architecture, watershed ecology, and regional sociology. These, and other, disciplines began to capture disciplinary understandings as computer-based simulation models in the 1960s and 1970s. Models were typically developed by scientists as tools to test ideas and hypotheses. They were generally difficult to use, required expensive computer time, and were often brittle in their operation. But, in the hands of the scientist/developer the models could be applied to management challenges. Application of these models required that the problem be associated with a single issue because the models themselves were discipline centric.

In the 1980s, graphical user interfaces began to be attached to models. The Apple Macintosh and then Microsoft's Windows provided graphical user interfaces for the selection and execution of computer programs. The X-Windows environment was developed by the X Consortium, lead by M.I.T., for the Unix operating system. For each of these environments subroutine libraries (and, more recently, object libraries) were written to support the development of graphical user interfaces for the operating system and for emerging end-user programs. In the 1980s and 1990s, hydrologic flow and transport models were outfitted with new graphical user interfaces. Examples include the Army Corps of Engineer's Groundwater Modeling System, SurfaceWater Modeling System (SMS), and Watershed Modeling System (WMS) (http://chl.wes.army.mil/software/; Holland 1998), DHI Water and Environment's MIKE product line (http://www.dhi.dk/), the Modular Modeling System (http://www brr.cr.usgs.gov/projects/SW_precip_runoff/mms/) from the US Geological Survey, and the US Environmental Protection Agency's BASINS (Lahlou *et al.* 1998). Statistics packages used in the analysis and modelling of social systems, were released with graphical user interfaces. This trend has brought more stakeholders in contact with scientific and engineering data, and has therefore increased the knowledge level of citizens involved in watershed/landscape management decisions. Experts continue to have access to the most sophisticated analysis tools and extensive databases and continue to provide additional information to the decision processes.

Software has also been developed in recent years that combined two or more simulation models behind a common look and feel and in a manner that allowed the automatic sharing of common databases. Geographic information systems (GIS) technology often provides the framework for making science-oriented simulation models more useful in management contexts. The GIS is now accepted in management offices and user interfaces can be built using the same look and feel of the GIS to operate models. The GIS stores the system state information needed by spatially explicit simulation models and can accept output from the models for later display and analysis. Many legacy surface and subsurface hydrologic models have been linked to GIS (Wilson 1996). The Geographic Resources Analysis Support System (Westervelt *et al.* 1992) is an open software environment that has been connected to many hydrologic models. Scientists and programmers can use the application programmer interfaces of GRASS to directly establish a geographic area of interest and resolution and then access the GIS data layers needed by the hydrologic model. The model can then write results back out as GRASS data files for further GIS analysis and display. The National Center for Geographic Information Analysis (NCGIA) has sponsored four Workshop/Conferences between 1992 and 2000 to help bring simulation modelling to the planning and management communities (Goodchild *et al*. 1993, 1996; NCGIA 1996). These communities are still adopting GIS, and using that adoption as a platform, these conferences have promoted the integration of GIS and simulation modelling. Efforts across the world have been represented at these conferences.

Another trend of the 1980s that continues into today is the pairing of simulation models into a single programme (or set of programmes) that allows the original models to run in an integrated fashion. The reasons for pairing models is that often the processes formally captured in each model rely on system state information that is dynamically simulated in the other model. For example, a hydrologic simulation model relies on the land cover. A vegetation succession model that changes land cover is based in part on the available soil moisture. Clearly there are feedback loops between hydrologic and vegetation succession models. Flood events can impact human settlement patterns, and those patterns in turn affect flood events. Construction projects that protect one area can inadvertently change the severity of flooding upstream or downstream causing the development of new flood management projects. When feedback loops are formally captured between urban development models and hydrologic models, a more complete tool for evaluating resource management can be created.

There are many reports of linking two or more simulation models. For example, Sengupta *et al*. (2000) created a spatial decision support system by combining a GIS and the spatial models GEOLP and AGNPS behind a single GUI to evaluate policy alternatives in a watershed. Vizcaino (2000) created a Spatial Decision Support System (SDSS) using AGNPS and WATFLOOD, a flood forecast hydrological model. SWAT, a quasi-distributed watershed model, and MODFLOW, a fully distributed groundwater model, have been combined with a user interface to create SWATMOD and applied to a Kansas watershed to demonstrate improved public acceptance of an integrated model with a friendly user interface (Sophocleous & Perkins 2000; Sophocleous *et al*. 1999). Prato & Hajkowicz (1999) developed a spatially explicit decision support system that employs a multi-attribute decision-making model to help a property manager select a land and water resource management system (LWRMS). A very important application of spatially explicit multidisciplinary simulation modelling is the development of Total Maximum Daily Load (TMDL) plans. As the United States federal government works with states to address the goal of reducing non-point source pollution it becomes increasingly important to identify the contribution of pollutants of each field. That information must be linked with an analysis of the potential for profitably using that land.

Chen *et al*. (1999) developed a decision support system to calculate TMDLs of various pollutants in the Catawba River Basin in North and South Carolina. Line *et al*. (1997) combined a comprehensive water quality model (AGNPS) with a modern geographic information system (GRASS) to create WATERSHEDSS. WetScape (Meyer *et al*. 1995) helps to evaluate alternative resource management options with respect to water quality, hydrology, and water supplies. The Lake Okeechobee Agricultural Decision Support System (LOADSS) allows land-use planners to assign any of 100 land management practices to 8,000 agriculture fields for the purpose of comparing alternative plans with respect to non-point source contributions of pollutants to streams and rivers (Negahban *et al*. 1996). An optimization module helps select field land management practices that minimize pollution while maintaining economic viability. There are many other examples of linking software programs to develop spatial decision support systems (Fredericks & Labadie 1993; Bennett *et al*. 2000; Srinivasan & Engel 1994). These are only a few of the examples where significant time and effort were invested to complete feedback loops in a modelled system by integrating two or more simulation models.

Another recent trend is the deployment of software simulation capabilities through Internet-based user interfaces. Leading GIS vendors including MapInfo (http://www.mapinfo.com) and ESRI (http://www.esri.com) now provide increasingly sophisticated software to support Internet-based GIS. This approach avoids the expensive tasks involved with packaging software, installing software, and developing extensive documentation for managing the software. Trame *et al*. (1997) describe the Fort Hood Avian Simulation Model, a land management decision support system that allows a manager to drive a spatially explicit simulation model through a Web interface. Lovejoy *et al*. (1997) documents a Web-based decision support system that allows communities to evaluate trade-offs. This approach removes the requirement for the users to install software on their personal computers.

Currently, the development of models is resource intensive in terms of both time and the level of expertise required for said development and execution. But in the future, the development and application of these models must become inexpensive enough that local parks, small towns, rural counties, and groups of interested citizens will develop their own spatially explicit models. Dupont *et al*. (1998) explored the barriers that have limited the impact of computer-based decision support systems on land management decisions. They cite limitations in technology, data and the workings of the organization. A four-step approach was outlined to improve acceptance of the technology:

1. Managers, stakeholders and decision-makers must begin by clearly defining their project, goals and budget, and then decide whether to use an integrated watershed management approach or a more discrete approach.
2. Through communication, managers, stakeholders, and scientists choose the most appropriate digital support tool.
3. Development of a new tool or adaptation of an existing one must take place within the context of the agency's management structure.
4. The agency places the tool into operational use following an initial trial period.

This approach is in contrast to the approach of a scientific team delivering a completed tool after its development without ownership by the intended recipient of the tool.

Watershed and landscape managers are responsible for managing the associated systems with respect to goals and objectives from a variety of stakeholders. On most landscapes there are competing interests that have conflicting goals and alternative management options are put forward to meet the goals. It becomes very important for the landscape manager to understand the implications of the proposed goals with respect to immediate, cumulative and long-term consequences. Today most land management decisions are made through the collective wisdom of long-term residents, scientists, citizens and politicians. Scientific simulation models (some spatially explicit) can be successfully employed by scientists to understand the implications of alternative actions when there is a single overriding objective. However, such models become more difficult to apply when there are multiple objectives, when the management objectives involve complex feedback loops among components of the system understood by different scientific disciplines, or when the stakeholder interest is very high. Any computational system that would seek to support landscape decision-making must provide a scientifically based capability that allows for the rapid (and inexpensive) development of multidisciplinary and collaborative models.

## FUNDAMENTAL GOAL

The background described above supports the need for development of a general purpose watershed/landscape simulation modelling environment that allows for the rapid development of locally specific simulation models to test proposed urban, watershed, and landscape management alternatives. Wilson & Droste (2000) outlined the needs for a contemporary Watershed Management Decision Support System (WMDSS). Based on those needs they recommend the development of a system that combines a model-base management system (MBMS), a database management system (DBMS) and a knowledge-base management system (KBMS). Behind each of these are simulation models, historic and current data, and human-based guidelines, desires, laws and requirements, respectively. In front of the management systems is a user interface that includes report generators and graphical views. A look at the desired characteristics of such a system and the challenges to the creation of the system will lead to design goals that will include a conceptual user interface. A fundamental goal in our development of the Land Management System (LMS) is to facilitate the inexpensive and rapid development of locally specific simulation models that can be used in an integrated manner to test the consequences and risks associated with proposed management strategies across watershed and/or landscape scales.

There are several key design objectives that must be realized before inexpensive and useful models of natural and human processes can be developed. End-users will range from decision makers and stakeholders who use modelling results to scientists and engineers who are involved in model building and/or model operation. Such a broad range of users requires highly flexible graphical user environments that support three levels: (1) extension of the modelling environment, (2) development of location- and application-specific decision support systems, and (3) model operation. Perhaps the best example of a user interface for model operation is found in the world of computer games. Maxis Software developed SimCity, a simulation based game that captured interactions between various modules of a city including economics, land-use, traffic, crime, tax revenues and happiness (http://simcity.ea.com/us/guide/). The user interface for running the simulation is intuitive and quick to learn despite the multidisciplinary nature of SimCity. A city runs in simulation time that can be adjusted faster or slower. The state of the city is continually updated during simulation. The user (player) is allowed to adjust system parameters during the simulation with consequences of those adjustments forming the feedback of the game. Behind the attractive interface are equations that dynamically update the state of the city based on the state of the city in the preceding time step. SimCity is but one of many examples of excellent user interfaces designed for the target end user. There are also a wide variety of excellent software development environments that support the needs of computer scientists. Graphical interfaces allow programmers to see their code in many different ways, select and incorporate code (objects) from organized libraries, and test/debug the code through efficient visual interfaces. Therefore, excellent examples of user interfaces are available for two of the three user levels (1 and 3). Level 2 provides challenges addressed in this paper.

The Level 2 user will assemble generic software objects or components that have been developed by software engineers and domain-specific scientist to create location- and application-specific models intended to test and evaluate alternative watershed and landscape management scenarios and proposals. Decision makers that include citizen stakeholders, politicians and planners will use these models directly or indirectly. The interface must provide access to a large set of natural, landscape, watershed and human urban objects that can be assembled to represent the landscape/watershed system being modeled or assessed. The individuals assembling these objects need not be computer programmers or scientists. Therefore, the objects must reflect commonplace real-world objects that can be placed on a map of the system. Available objects might include ‘neighborhoods’, ‘road’, ‘factory’, ‘lake’, ‘river’, ‘forest’ and ‘crop field’. These objects, and others like them, are associated with behaviours in time and space and interact with other objects in the system through those behaviours. Each object is associated with specific parameters that the modeller can set through easy-to-understand interfaces. For example, the ’neighborhood’ object might be associated with exact size and location, number of households, characteristics of households, demographics and associated vehicles. A ‘road’ is associated with engineering characteristics, width, speed, a safety index and traffic control.

Some of the objects available for the conceptualization should also allow modellers to visualize, control and interact with the model. Visualization objects (which access common LMS visualization tools used on each of the four levels of the system) provide monitors and probes that display the state of the system during a simulation run. Some of the displayable information can also be stored for later analysis. Further, the conceptual interface should result in models/modules being easier to assemble, operate and analyse/evaluate. Such an assembly would require participation by scientists and/or programmers for model calibration and verification, but once they are calibrated/verified non-modellers (decision makers, managers, stakeholders) should be able to make use of these models without a specific requirement for modeller/programmer involvement.

## THE LAND MANAGEMENT SYSTEM

The United States Army Corps of Engineers is actively pursuing the development of a next generation simulation

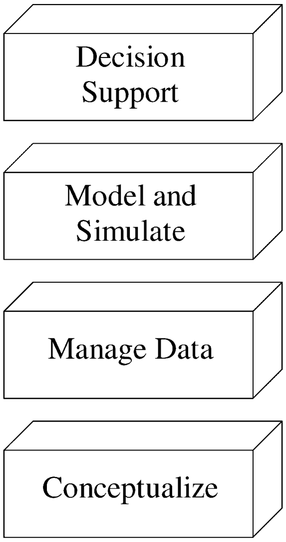


**Figure 2** | LMS functional levels.

modelling-based capability to evaluate alternative land and water management options. Initial design documents portray LMS as a four-tiered system (Goran *et al.* 1999; Holland & Goran 1999). These four tiers have been refocused recently to include those shown in Figure 2.

An overarching graphical environment is the entry point to all LMS services. Key documented design goals include:

- web-based,
- single, consistent look and feel,
- Microsoft Windows and Linux based,
- network and local applications,
- developed with marketplace standards (COTS browsers, Java, Windows, etc.),
- connection to standard land management decision support systems,
- state-of-the-art visualization,
- economic and risk analysis,
- provide what-if analyses.

The 'Decision Support' level provides stakeholders and managers with the ability to evaluate the tradeoffs between different proposed alternatives for meeting specified resource management goals. The 'Manage Data' level provides all the tools for seamless access to differing databases across networked and web-based environments. The 'Model and Simulate' level provides users with the ability to construct, set up, calibrate, verify and execute location-specific models from libraries of LMS model development modules. Finally the 'Conceptualize' level empowers users to specify, based on problem, location, goals, objectives, critical system components, which models to apply, and how to inter-connect them. Access to these different levels must be available through the conceptual user interface. Our challenge is to define a conceptual user interface that allows teams of scientists to rapidly create location-specific models and assessment tools. Meeting this need will help make spatially explicit simulation modelling as accessible and ubiquitous as today's geographic information systems.

The technical design of LMS fully embraces the goal of adapting legacy software. A Common Delivery Framework (CDF) has been established that makes it possible to establish services available on the Internet that run legacy software on machines remote to the end user. A user interface running on a decision-maker's computer might be designed to evaluate the potential social, economic, ecologic, agronomic and/or hydrologic consequences of alternative land management options. The user need not be concerned with the fact that some of the necessary analyses are accomplished through requests made through the CDF to a variety of legacy (and/or modern) programmes running on remote computers—even powerful supercomputers. Development of the end-user system requires a conceptual user interface that allows the model developer to combine and link the available local and remote LMS modules.

## PROPOSED CONCEPTUAL USER INTERFACE

The fundamental capability of LMS is to create and run location-specific simulation models in support of landscape/watershed decision support. These models must simulate the landscape/watershed processes of importance to the given location with respect to space and time. The LMS must have a conceptual interface that allows end users of the system (e.g. decision makers, policymakers and watershed managers) to specify the key components of the system being managed and to employ representative models of this system in a manner that is straightforward and natural. The learning curve is minimized by reflecting the system being managed in terms and approaches already familiar to the user while hiding or disguising viewpoints of software engineers, software languages and computational approaches and standards. An initial premise in our development is that it is impossible to pre-construct a single model that is applicable to all landscapes/watersheds. However, many of the parts of landscapes and watersheds are shared among systems. As an example, although urban and complex rural watersheds are composed of different parts, they share the same concept of a stream or receiving water. Therefore, it is reasonable to provide end users with *collections of components that might be part of their system*. And, because it is important to minimize any learning curve, *these components must reflect the general concept of real landscape objects*. They also must be organized conceptually the same way the user thinks of the world. For example, consider the development of a model of a watershed. The watershed is, in a common sense way, composed of a number of significant things like topography, streams, lakes, cities, towns, forest, farm, groundwater and weather/climate. The LMS environment will be straightforward to use if the objects available for constructing a watershed are formulated in an analogous manner. Each of these objects is conceptually composed of smaller objects and it is therefore important to allow model builders to provide a next level of detail by specifying the objects within each of the larger main objects. As an illustration, a farm is composed of fields, roads, structures, equipment, a management history and a management plan. So, the *objects must be hierarchically arranged—each potentially composed of sets of objects*. This approach is used for the commercial modelling environment Extend (http://www.imaginethatinc.com/).

There are five basic types of modules required in the LMS conceptual interface: *formulation, simulation, initialization, visualization and control*. The highest level of these modules would reflect common objects in the real world as understood by most people. That is, there may be 'dam', 'stream segment', 'road', 'neighbourhood', 'farm field', 'farm', 'forest', 'lake' and other common objects we see in our landscapes. The model development user interface will aid modellers, managers and stakeholders in identifying and establishing location-specific conditions, problems and potential solutions that are viewed as central to resource management. The simulation modules capture the understanding of how landscape/watershed components behave temporally and/or spatially. These modules may range from new, fully object-oriented developments to 'wrappered' versions of existing legacy models. Initialization modules contain system state information that is provided to the simulation modules to initialize them before a simulation run. Typically these will contain information commonly found in GIS and geospatial databases. Visualization modules will primarily provide windows into the outputs of a simulation during (or after) a model execution and may provide a wide variety of methods for inspecting the dynamic state of the model. These will also support the ability to store system state information into a variety of formats for later analysis and inspection. Finally, control objects will accept inputs (either from users or other models/data sources) during simulation runs that will be integrated into the simulation. For example instead of using a simulation module that generates recreational use of a lake, a control module might capture run-time decisions made by a human being during a simulation. Numerous control modules would allow control of visualization, storage of model outputs and control of model operation.

Watershed and landscape simulation models tend to require significant computational power, can be complicated in their setup and management, and can require software environments unavailable to users. Therefore, Internet browser-based interfaces can be important. Voinov & Costanza (1999) demonstrate a Patuxent River watershed simulation model available to users via the web that alleviates the need to require users to download and install software and for developers to ensure that that software will indeed run on a number of different platforms. The potential complexity of a watershed model, and the need for many collaborating individuals to work with a common model, suggests that the LMS needs to provide user interfaces via the internet that access and run remote simulation models.

Because modellers, managers and stakeholders are all aware of the importance of the spatial arrangement of system components, the *map is an important aspect of a conceptual user interface*. Readily available mass-marketed software, that helps users design rooms, layout gardens, and create houses, adopt this approach. The LMS conceptual user interface must allow users to place selected real-world simulation objects into modelling space. No commercial model construction software currently provides this capability. The Corps of Engineers' Hydrologic Modeling System (http://www.hec.usace.army.mil) (HEC 1998) uses an interface that allows modellers to construct a watershed by spatially arranging icons that represent streams, lakes, overland flow, etc., within a digital map.

As objects are placed spatially into a system being formulated, the *objects should automatically seek to establish connections with other objects in that system*. Automatic connections allow the user to develop new models without the tedious requirement of making obvious connections. However, any connections established must be visible and editable by the modeller. Similarly, *objects seeking connections with other objects must have access to the libraries of potential objects so that they might recommend object use to the modeller*. Unmet object connection requirements and connection recommendations must be readily viewable to the user as models are assembled. The type and number of potential objects recommended for possible connection would be a function of the scope of the location-specific problem as laid out during formulation.

The *initialization of the state of objects in a developed model must be automatic* and must be accomplished through interaction with geospatial systems and associated data. Conceptually, the system objects interact with one another during simulation time, but during the pre-simulation phase their state is established through information exchange with system state data that is often stored as geospatial data. The conceptual user interface must show both connections for initialization purposes as well as connections that provide run-time feedback loops.

As a system simulation proceeds, it is important to view and/or capture system state information. The conceptual user interface should employ the concept of *a probe*. A probe is an object that interrogates or polls a part of the system to report back information about the status of the system. Spatially explicit watershed and landscape simulation models have a tremendous amount of maintained system state information. Often it is not reasonable or practical to visualize or save the entire history of a simulation run. In such cases, model probes allow a user to select specific system state information for run-time display and for system state storage.

The interface must allow *users to easily locate useful objects in local and remote databases*. With an open system architecture there can be many dozens of LMS object libraries containing hundreds of potentially useful objects. These objects must be organized in a fashion that allows model builders to rapidly locate and use objects needed for a particular model. This requires that object builders adhere to model construction standards and conventions. For example data exchange formats must be well known and adopted. Data units must be accepted standards.

*Users must be able to create and modify objects*. Regardless of the depth and completeness of simulation modelling objects available, it will always be necessary for most users to modify the available objects and create new ones. In a gross sense, objects are either composed of computer instructions or other objects. At the foundation of all objects one will find computer instructions. These instructions will take many forms as there are many useful languages available to build LMS objects. However, it will be important to offer very simple object building environments for the modeller who wants or needs to construct a new model. The success of the commercial modelling systems such as Stella by High Performance Systems (http://www.hps-inc.com/; High Performance Systems 1997) and Powersim (http://www.powersim.com) suggest that their conceptual approach to model specification should be closely evaluated. For other users, access to more open-ended simulation languages will be important. Starlogo, a spatially explicit simulation modelling tool developed at the Massachusetts Institute of Technology (MIT Media Lab 1997), and Extend, a commercial simulation modelling package, both offer powerful text-based modelling languages to their users. Starlogo users develop models with the language. The Extend modelling language allows users to build new modules from scratch and/or through the combination of existing modules.

Modules can only be components of a simulation model if *common definitions are adopted for sharing system state information through module runtime inputs and outputs*. These definitions must specify units, error and uncertainty information, and time and space resolutions (where appropriate). Modules proposed as new additions to the system must be evaluated with respect to their adoption and/or formal extension of the data definition conventions.

The conceptual user interface must not bother the user with the computer science or scientific model details, but this information must be available upon request. The conceptual user interface of an automobile provides important access to steering, acceleration, braking, lights and signalling. Choosing to look a little deeper, the user can get information about the fuel level and will be warned about such things as low oil and status of the various systems. The user will also have access, with some extra effort, to the vehicle's systems. Special-order manuals help to provide the necessary expertise to work on, maintain and even modify these systems. The LMS system similarly should not concern the user up front with many system details. This includes information about required computational resources, software languages, network requirements and operational details.

The system must allow the user to lead the design, development and operation of the model while providing support expertise. During design and development of a new model, the system can continually analyse the developing system to recommend the use of available objects based on the input/output requirements of the objects already in the model and those in available libraries. Warnings associated with potential incompatibilities among connected objects can be generated based on object metadata analysis.

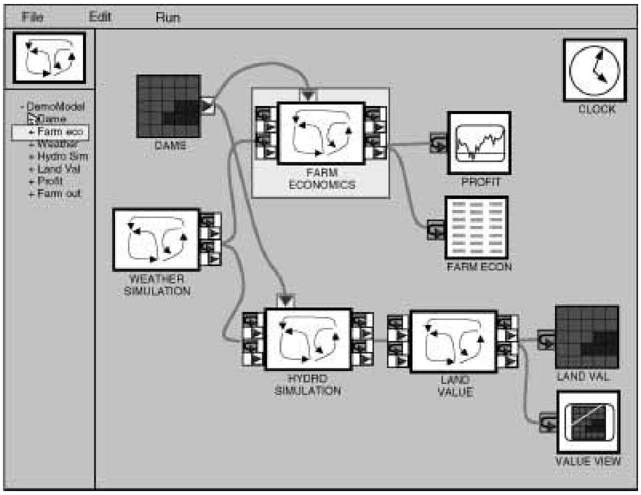


**Figure 3** | Icon-based graphical user interface.

## USER INTERFACE EXAMPLES

The system model developer will have access in LMS to sets or libraries of building blocks (modules) that can be assembled to reflect the system being managed, the inputs to that system, visualization requirements and analysis needs. There are two different general approaches that allow model developers to assemble simulation, initialization, visualization and control objects. Many modelling systems represent their objects with graphical icons. Icons are connected with lines to represent the exchange of information. Examples include Khoros (http://www.khoral.com/), ESRI's Model Builder, the Modular Modeling System, Stella, Extend and PowerSim. This approach is very useful and should be optionally available within the LMS conceptual user interface. A graphical depiction of this approach is provided in Figure 3. This approach becomes inadequate when the number of modules becomes large and/or there are many connections among the modules. In this situation it becomes more useful to use the second, a tabular, approach. Figure 4 provides an example. Model objects are listed at the tops of columns in an array. Shared variables are listed as row labels. At row–column intersections the words 'IN' and 'OUT' indicate if the variable is needed as an input to a module or provided as an output. As models are developed, this table is maintained automatically, leaving the modeller to focus on the gross model requirements. The model development environment graphically flags required variables for which there is no input. Also, the

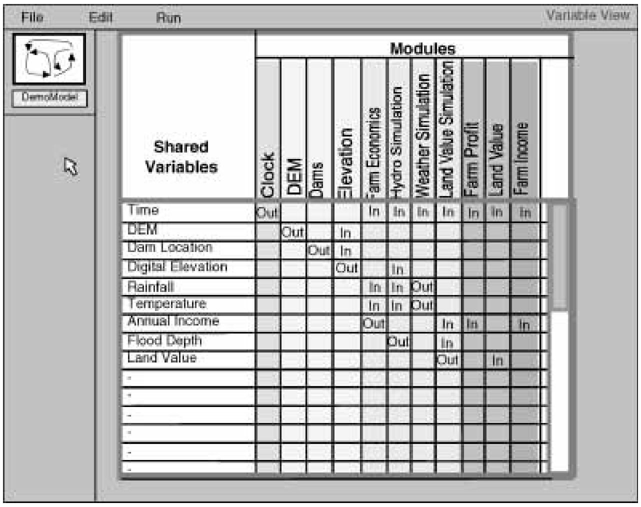

**Figure 4** | Table-based graphical user interface.

system will provide other information on request such as definitions of variables and suggestions for including modules that satisfy unmet variable requirements. The LMS conceptual user interface for model development must offer the graphical visualization for simpler models and the tabular interface for more complex models.

A geography-oriented interface is the third view needed for the LMS conceptual interface for model development. This view allows those familiar with processes associated with geographic locations to place and connect icons on a map image. This approach is use, for example, in the MIKE BASINS system and the Watershed Management System.

## REVIEW OF CURRENT INTEGRATION EFFORTS

There are a number of leading efforts that address the challenge of creating a system within which sets of objects representing the real world can interact. Several of these are briefly reviewed and then compared below.

The FRAMES (Framework for Risk Analysis in Multimedia Environmental Systems) system allows for the evaluation of risks associated with a pollutant moving sequentially through different media (Whelan *et al.* 1997). A graphical user interface allows a user to link various pollutant fate and effects models (air, soil, groundwater, exposure, intake and health impacts) together to holistically represent the movement of a pollutant from source to people. Each model reads required specification files and writes other specification files for input into other models. Communication between the models is facilitated in this manner – there is no need to reformat model output files into downstream input models. The simulation modules have been carefully developed to communicate directly through this process. The graphical user interface allows for the connection of the modules to meet the particular needs of the user. Modules can also be parametrized through the user interface. Specified system models can be saved for later additional development or operation. FRAMES does require that executable models all reside on one machine, although it is possible for a proxy process to, in turn, execute a remote procedure.

The FRAMES user interface presents simulation models as icons that can be arranged graphically to indicate the sequence of execution. In this environment risks associated with release of pollutants are computed based on knowledge captured in models of the different media through which pollutants travel before affecting human health. Other systems that have adopted this type of interface include the image processing systems ERDAS (http://www.erdas.com/) and Khoros (http://www.khoral.com/), and the GIS systems ESRI's ModelBuilder (http://www.esri.com). As the number of modules increases and the need to capture feedback loops is introduced, this conceptual user interface can become unwieldy.

The Modular Modeling System (MMS) (Leavesley 1996; Leavesley *et al.* 1996) provides a conceptual interface similar to FRAMES, but there is an important additional fundamental difference: feedback loops are allowed. Because of this, the wiring diagrams can be significantly more involved. MMS does not associate a figure/icon with each library, but instead displays the name of the module in the workspace. Adding a module to the model being developed simply involves clicking and/or dragging the desired module into that workspace. The arrows in FRAMES are associated with data files that are output from upstream modules to downstream modules. The

MMS arrows are instead associated with the sharing of particular data streams. That is, if there are five outputs available from a given module, there are five separately available output streams. Connecting the modules therefore involves more arrows and can rapidly become overwhelming. This increases the flexibility and the challenge to adequate graphical display of the connections on the screen. Connecting the modules is accomplished automatically as they are added to the workspace. Each module is known to have certain input requirements and output opportunities called input and output slots. Slots are associated with shared keywords that are precisely defined to ensure that modules communicating via slots are doing so appropriately. Upon placing the first module into an open workspace, the colour of the module is red to indicate that it has unsatisfied inputs. Through the process of selecting modules, linking module input and output slots, and rearranging the icons it is easy to assemble modules into a model representing a particular landscape. This conceptual model can then be turned into an executable model by selecting the 'Build' option from the 'Model' pull-down menu. Each module is associated with a particular subroutine that is found in the module library. The build process pulls the various subroutines together along with standard MMS simulation model interfaces, data I/O and visualization routines to compile all of the parts together into a single executable programme. In FRAMES there is no compilation; all modules are separately running programmes. Communication between MMS modules is therefore very fast and efficient.

The Argonne National Laboratory developed the Dynamic Interactive Architecture System (DIAS 1995) for the development of multidisciplinary management-oriented simulation models. DIAS is conceptually similar to MMS with a few notable fundamental differences. First, it is written with an object-oriented software language. This allows the various modules to be more self-contained. Second, it facilitates the execution of remote processes as part of the modelling and simulation. To adapt a legacy simulation model as part of MMS it is necessary to incorporate the subroutines as part of a single compiled programme. DIAS provides the opportunity for an existing model to be captured as an individually running process that is potentially executing on a separate machine on a network. The legacy code is 'encapsulated' in DIAS related code that allows the main DIAS model to communicate during simulations with the legacy code.

The Spatial Modeling Environment (SME) (Maxwell & Costanza 1997a, b) marries simulation modelling software like Stella to a powerful simulation execution environment. SME facilitates the simultaneous execution of Stella-like models for each grid cell associated with a raster GIS database. State variables in the models are initialized using information in GIS data layers. Modellers are not expected to be software programmers; they are encouraged only to develop the Stella-like models that will be run in parallel—accommodating each patch in a watershed grid. SME models are written using a simple convention that allows a cell state variable to be a function of not only the variables associated with the current cell, but also the variables of neighbouring cells. The cell simulation specification models are translated by SME into a common Modular Modeling Language (MML). A library of such translated models can be built up and maintained for future use. To create a spatially explicit simulation model, the modeller identifies model components (modules) from their library, matches variables where appropriate, and translates the MML code into C++ using the SME code generators.

Other SME options allow for the integration of channel flow-process models and point models. SME-generated models can read and write various GIS data formats and tables of data in different formats, and can generate variable graphics and maps during simulation runs. Because SME is written in C++, it is possible for a software programmer to link SME code to other C++ based simulation models. SME relies on the graphical user interface of Stella for model development. It does provide a powerful graphical user interface for parametrization, configuration and visualization of the final model.

DHI Water & Environment, an independent, international consulting and research organization affiliated with the Danish Academy of Technical Sciences, offers commercial software supporting the simulation of water through the environment and supply chains. Two products, in particular, provide modelling environments useful for the management of watersheds and landscapes. MIKE SHE is an integrated ground and surface water

**Table 1** | Model environment comparison

| | FRAMES | MMS | SME | DIAS | MIKE BASIN |
|---|---|---|---|---|---|
| **End user level requirements** | | | | | |
| Feedback loops among components | No | YES | YES | YES | No |
| **Model builder requirements** | | | | | |
| Re al-world objects | No | ? | No | No | YES |
| Objects are hierarchical | No | No | No | No | No |
| Automatic object-linking | No | Yes | No | No | ? |
| Automatic object library search | No | ? | No | No | No |
| **Modules** | | | | | |
| Simulation modules | YES | YES | YES | YES | YES |
| Initialization modules | No | ? | No | No | No |
| Visualization modules | No | ? | No | No | No |
| Control modules | No | ? | No | No | No |
| Little or no programming | YES | YES | YES | No | YES |
| **Fundamental capabilities** | | | | | |
| Temporally explicit simulation modelling | YES | YES | YES | YES | YES |
| Spatially explicit simulation modelling | YES | YES | YES | YES | YES |
| Map-based | No | YES | YES | YES | YES |
| User ability to create new objects | No | ? | No | No | No |

model that also handles water quality (Refsgaard *et al.* 1999; Singha *et al.* 1999). Available modules that users can connect as required include saturated and unsaturated groundwater flow, surface water, streams, linear reservoir, advection/dispersion solute transport, particle tracking, adsorption/degradation, geochemistry, biological degradation, crop yield and nitrogen consumption, macro pore flow, soil erosion module and soil plant system simulation.

MIKE BASIN provides a conceptual graphical user interface that allows planners and hydrologists to combine icons representing stream segments, nodes (confluences), reservoirs and water extraction and injection points (DHI 2000; Kjeldsen & Rosbjerg 2001). Models representing an area of interest can be graphically combined using an image of the area as a backdrop. Neither of these systems is open to allow third-party development of modules. They provide a toolbox for representing the hydrology of the region and contain water quality and agronomic components.

Table 1 compares five spatially explicit simulation modelling environments with respect to the conceptual user interface goals presented above. Each intends to support the development of spatially explicit simulation models that can be operated by watershed and landscape managers as a decision support system. Each supports the development of application- or location-specific simulation models, but this process requires a significantly greater level of technical expertise and limits the development of models to those fortunate enough to have software programming expertise. The easiest environment for building new models is offered by MMS. Modellers are able to connect simulation modules graphically and easily initialize those models through graphical user interfaces. However, it may be cumbersome for legacy models to be properly recast in a manner that takes full advantage of MMS's modular formulation.

FRAMES also offers an easy-to-use graphical user interface that allows a sequence of process models to be connected. The primary limitation of FRAMES is the requirement that each process model runs without run-time interactions with other models/modules. However, this approach is adequate where there are no feedback loops. DIAS offers the ability to develop easy-to-operate models, but the development of those models requires the technical skills of a highly trained programmer. SME allows non-programmers to develop sophisticated models through the Stella model development interface. However, movement of Stella models into SME is not always a straightforward process because SME does not recognize all of Stella's operations. MIKE BASIN uses the look-and-feel user interface of ESRI's ArcView 3.2. Users build networks of streams by drawing lines on top of GIS images representing a landscape. The interface provides a way for a user to parametrize a model rather than a method for building a model from a library of modules.

All of the systems, except for MIKE BASINS, support the notion of maintaining libraries of simulation modules that can be linked to construct user-specific models. FRAMES has a fixed library of modules that are actually former, stand-alone simulation models and tools. SME allows users to develop modules through Stella and it is up to the user to manage any library of modules. DIAS supports the development of libraries of DIAS objects and will be accompanied by libraries of objects developed by a broad user community. MMS is module library oriented and comes with a growing library of objects. None of these systems supports the notion of objects or components specifically designed to support initialization, visualization or control, although these could easily be developed in DIAS, MMS and SME. Each system does, however, have the ability to probe, visualize and store system state information during simulation runs, and these capabilities are built into the core of each system. Moving such functions to optional objects, modules or components would increase the alternatives available to system modellers.

Commercial simulation modelling tools are available to engineers and scientists and include systems like Stella and Extend. Neither of these systems provides spatially explicit simulation modelling capabilities, but their user interfaces deserve evaluation. Stella offers the modeller four basic icons that, when arranged according to user requirements, demonstrate a surprisingly powerful conceptual approach. The icons represents stocks/reservoirs, flows between reservoirs associated with valves, converters and arrows. Stella is a finite-difference simulation–modelling environment that requires modellers to specify algebraic and/or logical statements that change the system's state variables (reservoirs) from one time step to the next. The inter-reservoir flows are associated with valves; equations associated with a valve are used each time step to indicate the flow (numeric change) to and from associated reservoirs. Arrows from converters, valves and stocks to converters and valves indicate to the latter that the equation associated with the latter is a function of the former. Converters are simply valves that are not directly connected to stock flows and are used for intermediate calculations. This conceptual user interface does not provide any ability to create libraries of objects that reflect common real-world objects.

The Extend modelling system does provide and allow users to develop libraries of pre-defined objects. Each is associated with unique icons that can graphically depict the object and provide connection (input and output) points. Extend objects themselves may be an amalgamation of Extend objects making it possible to create a hierarchy of simulation objects.

## SUMMARY

United States Federal agencies including the Departments of Agriculture, Defense, Energy and Interior have been bringing more science into land management decision-making processes. Scientists have divided the complex landscape into components such as the social, ecological, economic, hydrologic and agronomic aspects so that each could be carefully understood. In many cases, the knowledge of system function has been captured as mathematical and computer simulation models. These have been scientific models developed by scientists to develop and test theories. In recent decades graphical user interfaces have made these discipline-specific models accessible to a larger audience. However, they are generally not useful for evaluating proposed land management strategies because they operate independently. Important feedback loops that intimately interconnect the components are not present in the discipline-centric models. Today these federal agencies are working to address this shortcoming of scientific models so that better next-generation land management models may be inexpensively constructed for individual watersheds. The US Army Corps of Engineers has embarked on the creation of the Land Management System (LMS), which will allow local policy makers, land and watershed managers, interested citizens and scientists to rapidly create models of managed landscapes from libraries of simulation modelling objects. This paper identifies the most important requirements of a conceptual user interface for the Land Management System. These requirements are derived from the need to make sure modelling and assessment results are easily accessible to decision-makers, managers and stakeholders. The basic design goal is to reflect the 'commonsense' real world through the LMS user interface.

## REFERENCES

Bennett, D. A., Sengupta, R., Beaulieu, J. R. & Kraft, S. E. 2000. Integrating simulation models and geographic information for environmental problem solving. In *Spatial Information for Land Use Management* (ed. M. J. Hill & R. J. Aspinall). Gordon and Breach, Reading.

Chen, C. W., Herr, J., Ziemelis, L., Goldstein, R. A. & Olmsted, L. 1999 Decision support system for total maximum daily load. *J. Environ. Engng.* **125**(7), 653–659.

DHI 2000 MIKE BASIN—a tool for river basin planning and management. *User Manual for MIKE BASIN*. Danish Hydraulic Institute, Horsholm, Denmark.

DIAS 1995 *The Dynamic Information Architecture System: A High Level Architecture for Modeling and Simulation*. Decision and Information Sciences Division, Argonne National Laboratory, Argonne, IL.

Dupont, J., Smitz, J., Rousseau, A. N., Mailhot, A. & Gangbazo, G. 1998 Use of digital decision support system tools for water management. *Rev. Sci. Eau/J. Wat. Sci.* **11**. 5–18.

Fredericks, J. W. & Labadie, J. W. 1993 Decision support system for conjunctive stream-aquifer management in the South Platte Basin. Inf. Ser. Colorado Water Resource Research Institute, 76.

Goodchild, M., Parks, B. & Steyaert, L. T. (eds) 1993 *Environmental Modeling with GIS*. Oxford University Press, Oxford.

Goodchild, M., Steyaert, L. T. & Parks, B. (eds) 1996 *GIS and Environmental Modeling*. GIS World, Fort Collins.

Goran, W. D., Holland, J. P., Barko, J. W. & Bruzewicz, A. J. 1999 *Plans for the Land Management System (LMS) Initiative*. 99/60, ERDC/CERL.

HEC 1998 *HEC-HMS, Hydrologic Modeling System User's Manual*. CPD-74, Hydrologic Engineering Center, Davis, CA.

High Performance Systems 1997 *STELLA: An introduction to Systems Thinking*. High Performance Systems, Inc; 45 Lyme Road, Suite 200, Hanover, NH.

Holland, J. P. 1998 Hydroinformatic advancements in integrated hydroenvironmental modeling systems. *Hydroinformatics '98*, IAHR, Copenhagen, Denmark.

Holland, J. P. & Goran, W. D. 1999 *Development of an Integrated Land System in Support of Department of Defense Land Management*. 99/86, ERDC/CERL.

Kjeldsen, T. R. & Rosbjerg, D. 2001 A framework for assessing the sustainability of a water resources system. *Regional Management of Water Resources, a symposium held during the Sixth IAHS Scientific Assembly. IAHS, Maastricht, The Netherlands* International Association of Hydrologic Sciences, Oxfordshire, pp. 107–113.

Lahlou, M., Shoemaker, L., Choudhury, S., Elmer, R., Hu, A., Manguerra, H. & Parker, A. 1998 *Better Assessment Science Integrating Point and Nonpoint Sources; Basins 2.0 User's Manual*. EPA-823-98-006, United States Environmental Protection Agency, Office of Water.

Leavesley, G. 1996 *The Modular Modeling System*. US Geological Survey, Denver, CO.

Leavesley, G. H., Restrepo, P. J., Markstrom, S. L., Dixon, M. & Stannard, L. G. 1996 *The Modular Modeling System—MMS: User's Manual*. 96-151, US Geological Survey, Denver, CO.

Line, D. E., Coffey, S. W. & Osmond, D. L. 1997 WATERSHEDSS GRASS-AGNPS model tool. *Trans. ASAE* **40**(4), 971–975.

Lovejoy, S. B., Lee, J. G., Randhir, T. O. & Engel, B. A. 1997 Research needs for water quality management in the 21st century: a spatial decision support system. *J. Soil Wat, Conserv.* **52**(1), 18–22.

Maxwell, T. & Costanza, R. 1997a *Distributed Modular Spatial Ecosystem Modeling*. http://kabir.umd.edu/SMP/MVD/

Maxwell, T. & Costanza, R. 1997b A language for modular spatio-temporal simulation. *Ecol. Model.* **103**(2–3), 105–113.

Meyer, S. P., Stiles, E. A. & Garcia, L. A. 1995 Demonstration of Wetscape: a terrain-based decision support system for planning wetland restoration and management' *Proceedings of Conserve '96*. American Water Works Association, Denver, CO, pp.93–97.

MIT Media Lab 1997 StarLogo. http://www.media.mit.edu/starlogo/

NCGIA 1996 *Proceedings, Third International Conference/ Workshop on Integrating GIS and Environmental Modeling*. National Center for Geographic Information and Analysis, Santa Fe, NM.

Negahban, B., Moss, C. B., Jones, J. W., Zhang, J., Boggess, W. G. & Campbell, K. L. 1996 Optimal field management for regional water quality planning. *Proceedings of Eco-Informa '96*. Global Networks for Environmental Information, p. 11.

Prato, T. & Hajkowicz, S. 1999 Selection and sustainability of land and water resource management systems. *J. Am. Wat. Res. Assoc.* **35**(4), 739–752.

Refsgaard, J. C., Thorsena, M., Jensena, J. B., Kleeschulteb, S. & Hansen, S. 1999 Large scale modelling of groundwater contamination from nitrate leaching. *J. Hydrol.* **221**(3–4), 117–140.

Sengupta, R., Bennett, D. A., Beaulieu, J. & Kraft, S. E. 2000 Evaluating the impact of policy-induced land use management practices on non-point source pollution using a spatial decision support system. *Water Int.* **25**(3), 437–445.

Singha, R., Subramaniana, K. & Refsgaard, J. C. 1999 Hydrological modelling of a small watershed using MIKE SHE for irrigation planning. *Agric. Wat. Mngnt.* **41**(3), 149–166.

Sophocleous, M. A., Koelliker, J. K., Govindaraju, R. S., Birdie, T., Ramireddygari, S. R. & Perkins, S. P. 1999 Integrated numerical modeling for basin-wide water management: the case of the Rattlesnake Creek basin in south-central Kansas. *J. Hydrol.* **214**, 1–4.

Sophocleous, M. & Perkins, S. P. 2000 Methodology and application of combined watershed and ground-water models in Kansas. *J. Hydrol.* **236**(3), 185–201.

Srinivasan, R. & Engel, B. A. 1994 A spatial decision support system for assessing agricultural nonpoint source pollution. *Wat. Res. Bull.* **30**(3), 441–452.

Trame, A.-M., Harper, S. J., Aycrigg, J. & Westervelt, J. 1997 *The Fort Hood Avian Simulation Model: A Dynamic Model of Ecological Influences on Two Endangered Species*. 97/88, Construction Engineering Research Laboratories.

Vizcaino, L. F. 2000 Integral system for nonpoint source pollution modeling in surface waters. *Diss. Abst. Int. Pt. B Sci. Eng.* **60**(7), 3432.

Voinov, A. & Costanza, R. 1999 Watershed management and the Web. *J. Environ. Mngnt.* **56**(4), 231–245.

Westervelt, J. D., Shapiro, M., Goran, W. D. & Gerdes, D. P. 1992 *Geographic Resources Analysis Support System (GRASS) Version 4.0 User's Reference Manual*. US Army Corps of Engineers Construction Engineering Research Laboratory.

Whelan, G., Castleton, K. J., Buck, J. W., Hoopes, B. L., Pelton, M. A., Strenge, D. L., Gelston, G. M. & Kickert, R. N. 1997 *Concepts of a Framework for Risk Analysis in Multimedia Environmental Systems*. Office of Research and Development, National Environmental Research Laboratory, US Environmental Protection Agency and Office of Environmental Management, US Department of Energy and Radiation Protection Division, Center for Risk Modeling and Emergency Response, US Environmental Protection Agency, Richland, WA.

Wilson, D. J. & Droste, R. L. 2000 Design considerations for watershed management decision support systems. *Wat. Qual. Res. J. Canada* **35**(2), 163–188.

Wilson, J. P. 1996 GIS-based land surface/subsurface modeling: new potential for new models. *Third International Conference/Workshop on Integrating GIS and Environmental Modeling*. CD-ROM. National Center for Geographic Information and Analysis, Santa Fe, NM.