

Linking local multimedia models in a spatially-distributed system

I. Miller, S. Knopf & R. Kossik
The GoldSim Technology Group, USA

Abstract

The development of spatially-distributed multimedia models has traditionally required a fixed geometric grid or mesh, or an *ad hoc* spatial representation for a compartment-type model. As geographic information system (GIS) applications have matured, an alternative approach is evolving where the spatial components of the model and the physical-processes components are developed separately. At run time the physical process models are mapped onto the spatial domain, the simulation model is constructed and evaluated, and results are exported to the GIS system.

In this approach one or more *generic* 'local' models are developed, then replicated across a spatial domain defined by a GIS system, connected to sources of spatial data, and finally the local models are interconnected to allow flows of information, physical media, and contaminants between the local models.

This approach is designed to allow multiple models to be instantiated and interconnected at appropriate locations. For example, models of physical processes such as rainfall and runoff (and seepage, erosion, etc.), could be linked to biotic process models such as crop growth, and to multimedia contaminant transport models.

The paper discusses alternative underlying conceptual approaches for this type of modelling, and presents the methodology selected for the GoldSim simulation software package.

1 Introduction

A dynamic simulation model is a computer-based model that simulates how a complex system will change over time. Simulation models are widely used in fields as diverse as biology, engineering, and business. However, most



simulation models available today are not designed to simulate geographically-distributed systems, and applying a general-purpose simulation model to something as complex as a geographic system is very difficult.

While there are numerous special-purpose dynamic spatial models which have been designed for specific applications, there is a growing need for flexible general-purpose dynamic spatial models that are capable of representing the interactions between many different entities or subsystems. Such models can be used to develop improved understanding and to support better decisions about these complex systems. In particular, multimedia transport models are often required to represent spatially complex multi-process systems.

A literature review was conducted by the authors, which found that a number of **custom models** of spatial dynamic processes have been developed. However, these tend to be of little value as **general-purpose models** of dynamic geographic systems.

Also, a few general-purpose modeling systems have been developed, such as the Spatial Modeling Environment (SMS) from the University of Maryland, the PCRaster system from Utrecht University, the Idrisi32 system from Clark Labs and the Integrated Dynamic Ecological Analysis System (IDEAS) from the University of Arizona. Westervelt and Shapiro [1] provide an excellent overview of alternative modeling concepts and systems that have been developed. A recent symposium on the topic (Maguire et al., [2], to be published in 2005) reviews a range of applications and issues.

Unfortunately, despite these primarily university-based model-development initiatives, dynamic GIS modeling is currently more of a concept than a practical reality. As a result, it has not been possible for us to identify or quantify different users and their requirements with any accuracy. In fact, the real applications for dynamic GIS will probably not develop until suitable systems are readily available.

There are significant existing communities that are involved in spatial-based modeling, but these are by and large not GIS specialists. Instead, they are the scientists and engineers who use finite element or other numerical models of processes such as groundwater flow and transport, atmospheric transport, river-basin flow, etc. Their models may import base data from GIS systems, but they are not directly linked with GIS systems.

We visualize two broad categories of potential applications for dynamic GIS models:

- Applications that essentially extrapolate some sorts of geographic processes over time: land-use projections, vegetation evolution, contaminant transport and other sorts of processes which are geographically complex but scientifically relatively simple; and
- Applications that address interactions between numerous processes and properties of geographic systems, which are too complex to model with 'pure' technical models.



2 Classification of dynamic GIS models

At a fundamental level, dynamic GIS models will be used to project future properties of GIS objects such as features, grid cells, or values in attribute tables. Because most geographic systems are heterogeneous, with several different kinds of components, it is envisaged that a dynamic GIS model will frequently consist of a number of different interacting sub-models, each modeling different processes, with the set of sub-models defined so as to cover a map region. For example, a model might integrate and link sub-models of terrestrial and aquatic ecosystems, rainfall/runoff, and erosion and transport processes. Note that several different sub-models might be active within the same geographic location, for example a rainfall/runoff and an ecosystem model which would coexist in the same GIS object.

The common denominator of dynamic GIS models, differentiating them from other types of dynamic models, is that the GIS models will automatically replicate behavior across specified classes of GIS objects. The process of replication could be as simple as applying a projected future time history to an entry in a value table (and thus to all map objects linked to that table row), or as complex as setting up multiple types of coupled nonlinear dynamic process sub-models over an entire map domain. For example, the value table projection might represent the increase in the average height of stands of fir trees, with a formula applied to predict the growth rate as a function of the current height. The complex model might represent all of the major physical and ecological systems operating in an entire watershed.

Where a particular type of sub-model is associated with a grid-cell or a feature, there will typically be some type of mask or attribute which would define which grid-cells or which features the model should apply to. For example, a model of processes in a coniferous forest would only be replicated for cells or polygons having coniferous forest present.

The following list describes five levels of increasingly complex dynamic GIS models. The simple models at the top of the list are easier to create and would in general be more frequently used than the models further down the list. The more complex model types at the bottom of the list are expensive to develop and require significant technical expertise by the modelers, but are potentially be of great value:

0. **Simple evolution models:** These simple models use a rule to project the evolution of a particular type of attribute over time, with no interaction with other attributes or with neighbors. The projected value could be a floating-point attribute of a specific type of grid-cell or feature (e.g. residual soil contamination levels following an accident), or an attribute value in a specific location in a table.
1. **Local dynamics models:** A local dynamics model specifies how multiple properties of a particular class of grid cell, feature, or table-row interact with each other to evolve over time. However, there are no interactions with neighbors. Multiple types of model



could be defined so as to ‘cover’ a region, so that every type of cell, feature, or unique table-entry would have its own model (e.g. a model for lakes, another model for fields, and another for forests).

2. **Coupled dynamics, single-system models:** In this category members of a set of cells or features interact with neighbors of the same model type to evolve over time. This allows one map region to influence another, to represent propagation (e.g. seeds, fire, urbanization) or transport processes (e.g. erosion, contaminant transport, water flow). However, all of the local models are of the same type and have the same variables. This approach requires the system to have topological knowledge of neighbors. The local systems defined on each map feature need to automatically identify their neighbors (if any) and to define appropriate linkages to transfer information with them. Also, in addition to specifying how to interact with neighbors, the user may have to specify what boundary condition to apply where there is no neighbor in any given direction (upstream/downstream; or north, northeast, east, southeast, etc.).
3. **Coupled dynamics, multiple sub-models:** These systems have multiple types of sub-models involved (e.g. models for lakes, fields, forests, rivers). In addition to the capabilities of the coupled single-system models, in these models each cell or feature has to identify if a different type of system is operating in its neighbors, and if so it has to set up appropriate interactions with them. There could also be a possibility of several different sub-models operating simultaneously within a single feature, e.g. biological and physical process models, with the user again having to specify the interactions between the models. For example, transport of contaminants through sediments and hydrologic processes could interact with models of vegetation growth and contaminant partitioning. Level 3 models will become even more complex where different types of GIS objects are involved in the different models. For example, if one sub-model is defined on a set of grid cells, and another is defined on a river network that runs across the same domain as the grid, the topological relationships necessary to define the interactions between the two model domains may not be readily available in the GIS system.
4. **Dynamically changing model structure:** In this most complex class of model significant dynamic changes can occur not just to the properties of model elements, but also to the model structure itself, such as:
 - a. Old features may be removed, or new features may come into existence. Automatically restructuring the model when this occurs will require disconnecting and disposing of old model elements, or instantiating and initializing new elements and their coupling to neighbors.

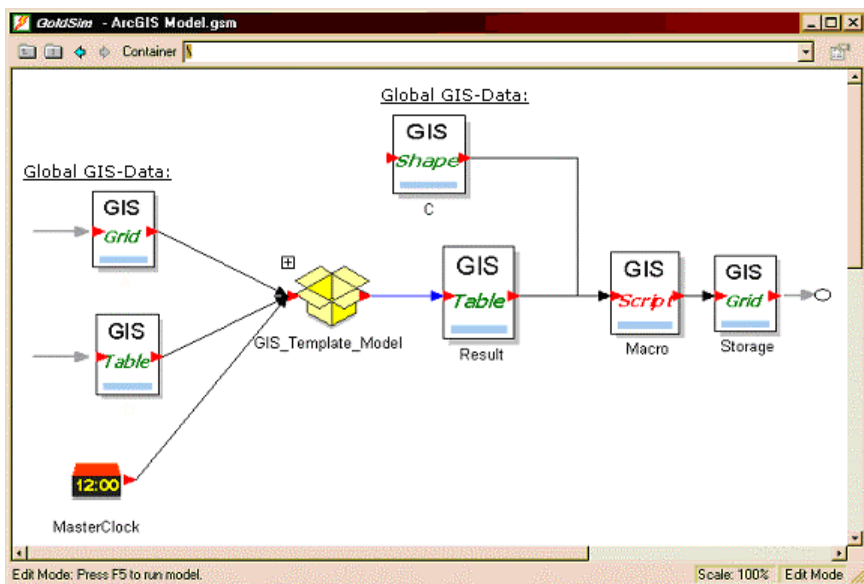


- b. The type of system operating in a given cell or feature might change, for example to represent conversion of land from one use to another. This is equivalent to removing the old local model and replacing it with a new one, though in this case it may be desired to inherit some properties of the old model in the new model.
- c. A feature's geometry might change with time, for example to simulate transient flooding. This type of moving-boundary problem can be difficult if the boundaries are sharp, which can require adaptive local spatial discretization in order to compute the boundary location accurately over time.

The ability to represent these sorts of dynamic changes to a system will require sophistication both on the part of the user and on the part of the simulation system.

3 Architectural alternatives

The authors have developed a prototype of a general-purpose dynamic modeling system, one that is capable of modeling quite general processes including multimedia contaminant transport. This prototype has been developed as a module for the GoldSim[®] software system, which is used for environmental and other modeling purposes.



From the user's point of view, there are two fundamentally different ways in which a simulation system such as GoldSim could be integrated with a GIS system. We refer to these approaches as the 'Simulator-Centric' and the 'GIS-centric' approaches.

In the Simulator-Centric approach, which was chosen by the authors, the simulation software provides the interface and data access tools for creating and carrying out the entire model simulation.

In order to gain access to GIS data, specific GIS components are used. These components provide (user) interfaces and functions to browse, select, import and present any type of GIS data (databases and files; different data types, e.g. grid, shape, line, etc.).

4 Central concepts of the GoldSim approach

The approach developed for the GoldSim software revolves around a concept referred to as replication. There are four central concepts for replication:

1. **Locations** that:
 - Represent a specific spatial area on a map,
 - Typically have some local properties (data) that are constant and are downloaded from the GIS, and
 - Have one or more processes that are to be modeled using local GoldSim models. The state of each process as it evolves through time is represented by outputs of the local models.

2. **Template models** that:
 - Define the properties and dynamic behavior of a system or subsystem at a generic map Location,
 - Have user-defined rules that specify which map Locations they are to be replicated into, and
 - When replicated across a Map, interact with each other and with the GoldSim model as a whole.

3. **A map** (a new GoldSim element type) that can be linked to a GIS and defines:
 - A spatial pattern of Locations,
 - Coordinates of the map Locations, and
 - Local data that can be used to:
 - i. Establish whether a particular type of template model is to be replicated as a local model at a Location, and
 - ii. Provide data inputs to the local model(s) at each Location.

4. **Connections** that:
 - Are automatically created as part of the replication process, and



- Are intelligent, in that the creation of connections between different local models can be conditional on the types of the models and on their topological relationships. For example, a connection representing soil erosive transport could be automatically created between local landscape models in adjacent Locations, but only where the first Location's elevation was greater than the second Location's elevation.

5 Technical issues

There are a number of technical issues that have to be addressed in order to create a functional GIS-linked multimedia transport model, including the following:

5.1 Accuracy and stability requirements

In general, dynamic simulation models become inaccurate if they use a time step that is too long. Intelligent, adaptive model time steps that guarantee accurate results with no user intervention are required, and it may be necessary to use different time steps for different subsystems or locations.

5.2 Result capture requirements

The user must be able to identify what calculated results need to be stored, and what can be discarded. Dynamic models generate large quantities of intermediate results that do not need to be saved, and in fact may be too large to save.

5.3 Time-based data requirement

Dynamic models produce time sequences of results, which will typically be expressed in two ways, as longitudinal time-histories of individual values (i.e. a history-plot of the projected population density in a particular urban area), and as 'Snapshots' of the entire system at a number of specific points in time (i.e. movies illustrating the system response over time). The GIS data architectures involved need the ability to access time-based data in these ways.

5.4 Category modelling requirement

Dynamic models in general deal with quantities and with intensive properties of system components: temperatures, densities, fluxes, etc. However, GIS systems frequently assign geographic components to categories (urban, agricultural, forested...), so an ability to model how components in a geographic system evolve from one category to another one may be required. This implies that in addition to the category assignment, additional properties of the system need to be simulated, in order to ascertain when a given entity will change its category,



and what the new category will be. This sort of logic is not typically employed in dynamic models.

5.5 Calibration requirements

Many dynamic models require adjustments of input parameters so as to provide a best fit to recorded, historical data before they are ready to use for predictive purposes. This process of model calibration (“inverse modeling”) can be extremely difficult for multi-variate, nonlinear dynamic models. Single simulations of dynamic models can be demanding in terms of computer resources and run-times, and trial-and-error calibration involving numerous simulations of a large model can be challenging. Thus, there is a requirement for some level of support for model calibration. Ideally, the software would incorporate a search/optimization capability for automated model calibration.

5.6 Uncertainty analysis requirements

Many environmental systems are subject to stochastic uncertainty due to weather/climate variations. Other common sources of uncertainty include parameter uncertainty (e.g. not being sure of a parameter, such as an erosion rate or a cost factor), and the chance of random events occurring. Random fields such as the distribution of subsurface contamination represent another type of uncertain component.

Models which can address these types of uncertainty, such as GoldSim, have to do repeated simulations, typically in the order of a hundred of more simulations of a given model. While this type of analysis is very valuable for decision-support calculations, at this time it is probably not realistic to expect to carry out uncertainty analyses in dynamic GIS models. The reason is the extreme amount of computer resources that might be required. There are also software design issues that would need to be addressed, such as how to display probability distributions of the modeled quantities on a map basis.

5.7 Static vs mobile (agent-based) model entities

Most GIS databases represent the properties of geographic entities in an essentially static map, and the most common form of dynamic GIS models will simulate the evolution of those entities over time. However, a completely different type of model exists where autonomous or semi-autonomous objects (‘agents’) move through and interact with each other and with their landscape. These objects represent entities such as wildlife, vehicles, or military units.

Different modeling approaches could be used to simulate:

- Dynamic agents moving through a fixed, static landscape with defined properties,
- A dynamic landscape subjected to and interacting with dynamic agents.

In either case, the software architecture required for agent-based modeling is significantly different from that required for just modeling the evolution of geographic entities in a static map.



5.8 Three-dimensional models

A fraction of dynamic GIS models will need to represent systems with vertical spatial discretization and interaction, such as oceanic or atmospheric circulation models and models of vegetative cover, soil layers, and bedrock. These models will typically be an order of magnitude more demanding of computer resources than two-dimensional models.

6 Summary

With the continuing evolution of computer hardware and software it is becoming possible to model the behavior of complex coupled systems over geographical regions. At the heart of the approaches is the ability to separate the definitions of the different processes and how they are to be modeled from the topological, geometrical, and data attributes that can be stored in GIS systems. The simulation software then constructs the actual model system in real time, downloading GIS information in order to set up and model the overall system, and up-loading results to the GIS for storage and subsequent display.

References

- [1] Westervelt and Shapiro, 2000, Combining Scientific Models into Management Models, 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM\$), Banff, Alberta, September 2000.
- [2] Maguire, D., Goodchild, M., & Batty, M., GIS, Spatial Analysis and Modeling, ESRI Press, 2005 (In press).
- [3] Pullar, D. Embedding map algebra into a simulator for environmental modeling, 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM\$), Banff, Alberta, September 2000.

