International Congress on Environmental Modelling and Software

1st International Congress on Environmental Modelling and Software - Lugano, Switzerland - June 2002

Jul 1st, 12:00 AM

A Modelling Framework Incorporating a Map Algebra Programming Language

D. Pullar

A Modelling Framework Incorporating a Map Algebra Programming Language

D. Pullar

Geographical Sciences and Planning, The University of Queensland, Brisbane QLD 4072, Australia

Abstract: The paper presents a modelling framework to integrate dynamic analysis and spatial analysis, and to apply this to a hydrological application. The dynamic analysis runs a simulation of a set of process modules. Modules for spatial analysis functionality are implemented in software libraries, or components. Currently we support components for a map algebra language called MapScript, and for spreadsheet calculations using Excel. An important issue in environmental models and in any modelling framework is dealing with different scale processes. When processes occur over different time or spatial scales but are still integrally related, this becomes a problem for applying efficient modelling solutions. The paper reviews two common strategies for solving this problem, namely multistep and multigrid methods. Because of the heterogeneous nature of landscapes processes neither is seen as offering a satisfactory solution. We have tried a variant to a multistep solution by allowing the user to specify the timing of different processes that occur in a landscape model. The paper describes the modelling framework and language specification used to describe module execution. The specification is written in XML to leverage its widespread use in web computing environments. The modelling framework may be used for diverse modelling applications in hydrology, landscape assessment, geomorphology and ecology. The paper will demonstrate a mock hydrological application to model runoff in a small watershed.

Keywords: Simulation; Modelling framework; GIS

1. INTRODUCTION

Environmental models simulate changes in the physical landscape in space and time. A number of approaches have been put forward to perform environmental modelling in a computer [Ford, 1999]. Two common approaches are: i) to use a modelling language, or ii) to use a Geographical Information System (GIS). Most popular modelling languages like Stella or MATLAB do not include any spatial modelling capability. The other option is GIS which is traditionally known for its spatial data management and visualisation capability. A GIS may be linked to external modelling programs transferring data to and from the external model, and to display results in a map-based view [Westervelt and Shapiro, 2000]. The main disadvantage of this approach is that the GIS lacks any dynamic analysis capability [Burrough, 1998] so coordinating the execution of several modules or external programs is difficult. An alternative approach is to use a modelling framework that supports the coupling and coordination of modules [Pullar, 2000]. The modelling framework provides the dynamic modelling capability, and the GIS performs spatial analysis within a simulation environment. Several modern systems use this approach including IMA [Villa, 2001] and PCRaster [Wesseling et al., 1996].

This paper describes our experience with linking a GIS software library, henceforth called a component, into a modelling framework for spatial simulations. The GIS component we use is based upon a map algebra programming language [Pullar, 2001] with special operators and control constructs for dynamic modelling [Pullar, 2002]. The issue investigated in this paper is efficiently performing simulations when dealing with heterogeneous physical processes [Jensen and Mantoglou, 1993]. This problem arises because of spatial variation and differences in temporal scales for environmental processes. Landscape heterogeneity is an important topic in geography with related issues of scale and complexity [Phillips, 1999]. In a modelling framework the problem emerges as an issue of efficiency and stability of numerical solutions. The paper reviews a number of methods devised to deal with spatial-temporal variability in solving process equations. Our goal is to understand the general applicability of these approaches to spatial simulations. We hope to include these notions in the contextual and semantic specification for a simulation [Villa, 2001].

The outline of the paper is as follows. The next section discusses simulation and describes a language specification for our modelling framework. Sections 3 and 4 discuss spatial dynamics and review efficient numerical solutions. Section 5 describes the additions to a modelling framework specification for time scale issues, and illustrates this using a hydrological example for overland and channel flow. The conclusion suggests a simple approach to include multiple time steps in module simulations.

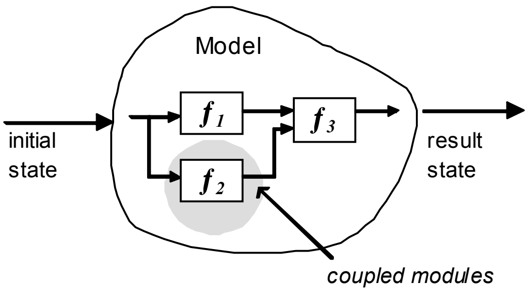

Figure 1. A model composed of a set of coupled modules. Modules implement their functionality using one or several software libraries, or components.

2. SIMULATION

This section discusses simulation and our use of a functional network as a modelling framework. Basic modelling formalisms have been described for discrete time, continuous and discrete event systems [Zeigler et al., 2000]. These differ by the way they handle dynamic changes in the model. Discrete time systems express state transitions through fixed time steps, continuous systems express changes through ordinary differential equations, and discrete event systems adopt a triggered execution mode where states change based upon previous states, conditions and inputs. Discrete time systems are very appropriate for handling spatial dynamics. This is because change in environmental systems may be expressed using a stepwise model of execution over discrete time and spatial intervals (cells).

Specifications for system models define the issues of timing and state transition for modules [Zeigler et al., 2000]. In our work we have developed a specification of a system model based upon a functional network. As shown in Figure 1, a system model is understood to consist of a set of coupled or linked modules. Modules are represented as nodes with arcs defining transition dependencies. Different types of modules may be supported using software libraries or components, e.g. Excel spreadsheets, MapScript, Stella. The specification includes the definition of modules, their topology for transitions and any common data parameters shared by modules. It can be thought of as defining all contextual parameters for dynamic aspects of the model and modules. Villa [2001] defines a number of criteria for model compatibility. Many of these criteria relate to space, time and behaviour of modules and their different modelling paradigms. We use a similar approach to define the context parameters for each module within a model, currently this is limited to: i) data parameters, ii) timing parameters, and iii) selections on spatial data. In the future we plan to add other semantic contexts.

The specification needs to be written in a language, we use eXtensible Markup Language (XML). A model is composed of a set of modules which correspond to different components is described in an XML document. It is organised hierarchically as a tree with cascading branches of nodes, each node is an XML element. The top level nodes include timing information, modules descriptions and network topology. See figure 2.

Timing defines the ordering for model execution. A universal clock controls the timing for execution of components, and handles various time units. A time unit, such as 1 second, 1 minute, or 1 year, is chosen to correspond to each computational step covering the interval Δt. Modules are executed for each computational step from a given start to a given end time. There are many issues related to timing and module execution, such as initialisation, relation to physical time, quantisation, and synchronization between processes. We support timing execution in three modes: an initialisation mode that runs before a simulation, a process mode that performs simulation, and a termination mode that runs after the simulation.

Modules define the processes that are executed. XML is used to specify the name of a module, its component type, data parameters, and its contents. It is desirable to support several types of components. Currently we support a spatial analysis tool called MapScript [Pullar, 2001], and spreadsheet processing using an Excel component. The content information for a module may be included in the description or in a named file. For instance a MapScript component includes data parameters and map algebra program code. An

Excel component includes data parameters as spreadsheet cell references and a named Excel spreadsheet file. XML is also used to specify the functional network of modules as a set of directed connections between named modules.

The model specification may be viewed in XML as a tree view with branches and nodes. Nodes can be expanded or collapsed, depending on whether or not the node has child nodes. Besides editing tasks the only commands are to load and execute an XML document. Examples presented later in this paper will show working models.

```
<MODEL NAME="Main">
  <TIMER START=1 END=10 STEP=1 UNIT="hours"/>
  <MODULE NAME="a" COMPONENT="Excel"
    FILENAME="calc.xls" TIME_REFERENCE="A1" />
    <DATA NAME="k" REFERENCE="B1" />
  </MODULE>
  <MODULE NAME="b" COMPONENT="MapScript">
    <SELECT="REGION=Pine" />
    <DATA NAME="k" />
    <CODE EXECUTION="Start" >
      … MapScript initialisation …
    </CODE>
    <CODE EXECUTION="Run" >
      … MapScript program simulation …
    </CODE>
    <CODE EXECUTION="End" >
      … MapScript termination …
    </CODE>
  </MODULE>
  <NETWORK>
   <CONNECTION NAME="a_b" FROM="a" TO="b"/>
  </NETWORK>
</MODEL>
```

Figure 2. Syntax for the specification language

3. DISCRETE SPATIAL DYNAMICS

Spatial simulation modelling generally covers two broad computational techniques: cellular automata and space-time differential equations. The approaches differ on the transition rules used to advance a simulation. Cellular automata use general rules uniformly applied over a grid to modify the local neighbourhood of a cell. Despite its simplicity it can achieve interesting patterns such as the Game of Life. Space-time differentials are expressed as difference equations in a discrete domain. Difference equations represent the change of state over time $\Delta x = x_{t+\Delta t} - x_t = rx_i$ where the rate of change r is related to the differential equation $dx/dt = rx$. Examples of the use of differential equations are found in hydrology [Maidment, 1996] [Mitas and Mitasova, 1998] and environmental engineering [Câmara et al., 1996].

The research described in this paper uses discrete space and discrete time for physical process models. Using a combination of map algebra and a modelling framework we have been able to address a large number of environmental problems. However problems do arise in using this simple computational approach. Physical processes occur at varying rates in reality. For instance in hydrology overland flow velocities vary significantly between steep and flat terrain. The extent to which one process can be separated and treated independently from others is related to issues of scale and the type of analysis being undertaken. We recognise two situations that arise for inter-related processes:

- *bounded* processes means that model parameters may be treated as constant and processes run independently from one another,
- *integral* processes means that model parameters change dynamically as each process runs.

An example of bounded processes occurs with long term soil-water budgets. It is possible to hold climatic conditions as constant when computing hydrological responses, and then afterwards use aggregated values for water surplus in a soil water budget. This can be done because precipitation events have a more refined temporal scale than climatic processes that affect soil-water storage. In general, processes acting over sufficiently different spatial or temporal scales can be considered independently of each other [Phillips, 1999]. As one would suspect integral processes occur simultaneously at similar spatial and temporal scales. For instance, surface runoff and stream flow are interrelated in this way as they occur during the course of a storm event.

In simulating a set of integral processes changes are propagated over each time step. The choice of numerical discretization is critical for both the spatial resolution and simulation time step. In many natural system processes there are large variations in levels of numerical discretization required to solve the differential equation formulations in a stable and accurate manner. Heterogeneity is a fundamental problem in process models like hydrology [Jensen and Mantoglou, 1993]. Landscape properties such as slope and surface cover are highly variable across space, and system drivers such as climate are highly variable over time. These have an important effect on flow and transport processes that influence patterns of water quantities and of materials concentrations across a landscape. Some modelling texts suggest finding an average time interval or spatial unit that delivers a stable solution. This involves identifying the most rapidly changing parameter, and then using a time step that is half the average time interval needed to model changes in that parameter. When models have parameters with short and long time intervals then this introduces serious numerical efficiency problems called 'stiff' equations by mathematicians [Hirsch, 1988]. Ford [1999] recommends other solutions such as using a steady state equivalent for rapidly changing parameters to capture their overall influence. In some cases this is a reasonable approach, but in many landscape models natural heterogeneity is caused by a spatial structure and integral processes that cannot be averaged out.

4. NUMERICAL APPROACHES

This section discusses some general numerical techniques to deal with heterogeneous processes. The techniques are more suited to *implicit* methods where a series of numerical differential equations over a grid are set up to solve a boundary-value problem, as opposed to *explicit* methods which calculate values over a grid using difference equations and values from previous time steps. Our interest is to appreciate physical conceptualisations of spatial dynamics and not to present detailed numerical approaches. In reviewing these methods we hope to recognize rules that can be applied broadly to modelling frameworks. Two common approaches for dealing with heterogeneous processes are:

- multistep approaches which use a variable time step in simulations, and
- multigrid approaches which use a variable grid resolution in simulations [Hirsch, 1988].

Many physical process models can be expressed as a series of ordinary differential equations which satisfy some initial conditions. A numerical differential equation solver is used to find an implicit solution based upon the process achieving a equilibrium or conserved state. Numerical schemes deal with the problem of efficiency and the solver converging to an accurate solution. The idea is to take big steps to get to a close answer and then smaller steps to find an exact answer. Multistep methods use a strategy to vary the size of the time step Δt and use intermediate results to converge to a solution (Figure 3). The solver monitors the accuracy of the solution changing the step size to larger steps when the computed state is varying slowly and to smaller time steps when the computed state varies rapidly.

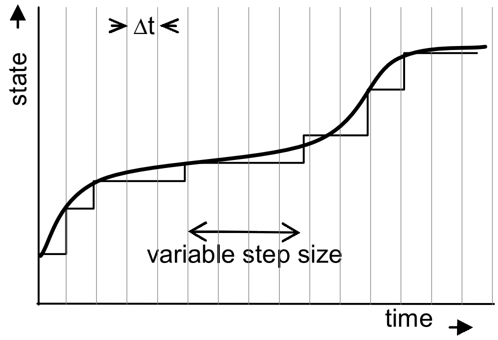

Figure 3. Variable time step or multistep strategy.

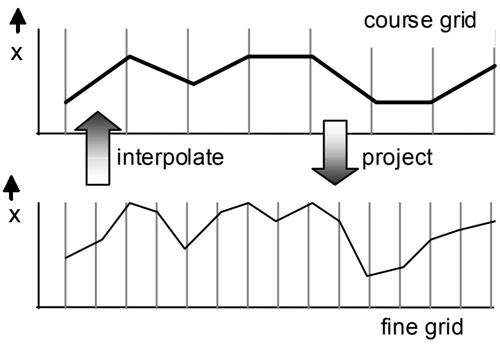

Figure 4. Transfering between coarse and fine grid for multigrid strategy.

Multigrid methods have a spatial context treating differential equations that are solved over a finite grid (Figure 4). For equations over large grids they use a strategy to vary the size of the grid between fine and coarse to converge to a solution. It may be viewed as inner and outer solvers with the inner solver performing local adjustments on a fine grid (relaxation technique) and the outer solver using a coarser grid providing an approximate guess (initial value) for coarse components of the solution. A number of approaches are used to transfer values between solver grids: coarsening (generalise) uses projection or averaging, and refining (densify) uses interpolation.

The multistep and multigrid methods give some insight for strategies that link process modules. In a GIS a raster grid is a common representation for environmental surface layers. The grid size or resolution is part of a raster layer and operations are available for resampling to different resolutions. Metadata about the process step size is not known. This is an important property that needs to be known by processing modules in a simulation. The next section describes how a sampling strategy has been incorporated into our modelling framework to capture the sampling time rate for certain spatial layers.

5. MODELLING EXAMPLE

This section discusses our approach to incorporate efficient dynamic processes in a modelling framework. As a simple example we will model terrain runoff with separate modules to represent shallow water flow across the landscape, and channelised flow within streams. The movement of water is described by the continuity equation:

$$\partial h/\partial t = e_t - \nabla\cdot q_t \qquad (1)$$

where h is the water height (m), e_t is the rainfall excess (m/s), q is the discharge (m/hr) at time t. Discharge is assumed to have steady uniform flow conditions, and is determined by Manning's equation:

$$q_t = v_t h_t = \frac{1}{n} h_t^{5/3} s^{1/2} \qquad (2)$$

where q_t is the net flux (m/s), h_t is water depth, and s is the surface slope (m/m). An explicit method of calculation is used to compute velocity and depth over raster cells, and equations are solved at each time step. A conservative form of a finite difference method solves for q_t in equation (1). Details of the method can be found in Julien *et al.* [1995].

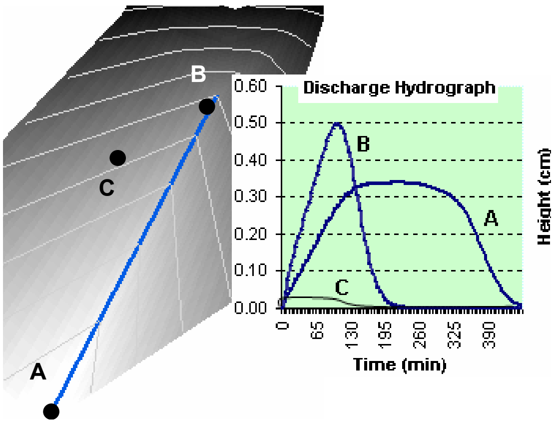

Figure 5. Mock watershed with output discharge hydrographs for channel (A,B) and terrain (C).

During a simulation water flows at much higher velocities in channels than overland terrain. Figure 5 presents an example of a storm simulation for a mock catchment showing how channels (points A and B) have much higher flow than the terrain (point C). One of the difficulties in applying a multigrid approach to this type of problem is that the channel needs to maintain a fine resolution so coarsening the grid size would only degrade the accuracy of the solution. A multistep approach is reasonable. When dealing with difference equations this requires the user to specify the time steps as part of the simulation.

```
<MODULE NAME="overland"
COMPONENT="MapScript"
STEP1=10>
<CODE EXECUTION="run>
<![CDATA[
doflow(dem)
   if (not channel)
     // compute flow velocity
     fvel =  1/n *  pow(depth,m) *
sqrt(grade)
     // compute net flux as gain -
loss
     l_loss = depth *  (fvel * step1) /
dx
     gain = sum(inflows() * l_loss)
```

Figure 6. XML module specifications for overland flow at a 10 minute time step and channel flow at a 1 minute time step.

In our approach separate modules with different time steps are used to model overland and channel flows. Figure 6 shows the module specification in XML. To accommodate the different time steps we specify attributes in XML for module timing and data sample time. A model includes timing parameters for *start*, *end* and a default *time step*. Each module is allowed to have separate specifications, but for computational convenience the simulator requires that time steps are even multiples of each other. For instance modules with time steps of 5 minutes, 10 minutes and 60 minutes are acceptable. This restriction could be removed at the expense of more complex simulation scheduling. Data parameters that have a temporal significance within a simulation are also given an attribute to indicate their temporal validity. The attribute is called *sample time* and is used to tell MapScript modules how to handle parameters for rates used in program code. This provides some level of user control for simulations, but obviously is far short of providing more adaptive control.

6. CONCLUSIONS

The paper has presented a modelling framework that includes dynamic analysis and spatial analysis. We use an approach where we embed GIS spatial analysis in a simulator to run environmental models. Users are given a high level language to write spatial models using a map algebra scripting language called MapScript. This is done in a modular way so each program is treated as a module in a larger simulation. A specification language is used to describe modules, their process order, and timing parameters. One of the more significant issues in running environmental models is dealing with heterogeneity in landscapes. Environmental models may include a number of integral processes that cause changes in state at different space-time intervals. This poses a significant problem for efficient numerical solutions. We have reviewed two commonly applied approaches for numerical solvers, namely multistep and multigrid methods. Neither method is seen as offering a generic solution. However, we have explored the applicability of allowing the user to specify timing constraints with the module specification, and providing some metadata about the sampling rate for time dependant spatial parameters. For instance, spatial data layers with distributed values for discharge rates and fluxes must include their temporal context within a process model. Within a simulation, modules are executed at each computational step where the user has control on the timing of execution. The user specifies the time step and the sampling time for any time related parameters. Using a simple hydrological example we demonstrate that is sufficient to deal with integral processes for overland flow and channel flow in a simulation. Future work will attempt to generalise this approach so it is more adaptive to the process. For instance, a condition to assess the stability of a solution could also be specified. In hydrological flows a velocity term is easily evaluated to test if an appropriate time and spatial step is being used. This can only be done dynamically as flow depth changes with each computation step. Future work will explore this and other options.

7. REFERENCES

Câmara, A., Ferreira, F. and P. Castro, Spatial Simulation Modelling. In M. Fisher, H. Scholten and D. Unwin (eds), *Spatial Analytical Perspectives on GIS*, Taylor & Francis, 1996.

Ford, A., *Modeling the Environment*. Island Press, Washington, 1999

Hirsch, C. Numerical Computations of Internal and External Flows, Vol.1 *Fundamentals of Numerical Discretization.* Wiley and Sons, Chichchester, 1988.

Jensen, K., and A. Mantoglou, Future of Distributed Modelling, In K. Beven and I. Moore (eds.), *Terrain Analysis and Distributed Modelling in Hydrology*, Wiley, 203-212, 1993.

Julien, P., Saghafian, B. and F. Ogden, Raster-Based Hydrologic Modelling of Spatially-Varied Surface Runoff. *Water Resources Bulletin* 31(3), 523-536, 1995

Maidment, D., Environmental Modeling Within GIS. In M. Goodchild, L. Steyaert, B. Parks B and C. Johnston (eds) *GIS and Environmental Modeling: Progress and Research Issues*, GIS World Books, 315-323, 1996

Mitas, L. and H. Mitasova, Distributed Soil Erosion Simulation for Effective Erosion Prevention. *Water Resources Research* 34(3), 505-516, 1998

Phillips, J., *Earth Surface Systems. Complexity, Order and Scale*. Blackwell, Oxford, 1999.

Pullar, D., Embedding Map Algebra Into A Simulator For Environmental Modeling. *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4) CD-ROM*, Banff, Canada, 2000.

Pullar, D., Map Algebra and Neighborhood Functions, *GeoInformatica* 5(2), 145-163, 2001

Pullar, D., MapScript: A Map Algebra Programming Language for Spatial Modelling. *Transactions in GIS*, SUBMITTED, 2002.

Tomlin, D., G*eographic Information Systems and Cartographic Modeling*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Villa, F., Integrating Modelling Architecture- A Deductive Framework for Multi-paradigm, Multi-scale Ecological Modelling. *Ecological Modelling* 137, 23-42, 2001.

Wesseling, C., G., Karssenberg, D., van Deursen, W. and P. Burrough, Integrating Dynamic Environmental Models in GIS: the Development of a Dynamic Modelling Language. *Transactions in GIS* 1, 40-48, 1996

Westervelt, J. and M. Shapiro, Combining Scientific Models into Management Models. *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4) CD-ROM*, Banff, Canada, 2000

Zeigler, B., Praehofer, H., and T. Kim, *Theory of Modeling and Simulation.* Academic Press, 2000.