

SpaSim: A Software to Simulate Cellular Automata Models

Niandry Moreno^{1,2}, Magdiel Ablan¹, Giorgio Tonella^{1,3}

¹ Centro de Simulación y Modelos (CESIMO), Facultad de Ingeniería

² Escuela Técnica Superior Forestal, Facultad de Ciencias Forestales y Ambientales, Universidad de Los Andes, Mérida, Venezuela

³ Facoltà di Scienze Economiche, Università della Svizzera italiana, Lugano, Switzerland
morenos@ing.ula.ve, mablan@ing.ula.ve, tonella@ieee.org

Abstract: Simulation models are a key tool to study the dynamics of ecological systems where space plays a fundamental role. These kinds of models in which the dynamic of the system is space-dependent are called spatial simulation models. The most frequently used approach to simulate spatial models consists of a simulation program (typically developed in a general purposed language) that periodically exports results to a Geographic Information System (GIS) to visualize or analyze the spatial component. This type of approach requires of a user programming skills and to be knowledgeable about GIS, limiting the use of the model and increasing the time necessary to develop and complete a project. A second approach combines in one interface the simulation program with the tools needed to manipulate the spatial component of the models. Although these types of systems are user-friendly, they are also less general since they deal with only a very specific kind of system. This paper presents the design and implementation of *SpaSim*, a software to simulate spatial models that overcome the disadvantages of these two approaches. The program allows the user to build, simulate and perform spatial and spatial-temporal analysis on the same environment. Currently, the software supports the development of models based on cellular automata. Demonstration of the software is given via implementation of a recently published model. *SpaSim* reduces considerably the time to develop cellular automata models and give users an appropriate combination of flexibility and ease of use.

Keywords: Spatio-Temporal models; Geographic Information Systems; Simulation Software; Cellular Automata

1. INTRODUCTION

Space plays a fundamental role in most ecological problems. For example, studies dealing with landscape forest modeling (e.g. [Acevedo et al. 1995]) or the occurrence and behavior of fire in a natural area (e.g. [Gardner et al. 2000]), are modeled taking in consideration the space that model variables occupy along a period of time. In this paper we present a software to solve spatial models, the type of ecological models where space is given explicit consideration.

The simulation of spatial models is called spatial simulation. A spatial simulation describes the changes in a spatial pattern from time t to time $t+m$, i.e.,

$$X_{t+m} = f(X_t, Y_t) \quad (1)$$

where X_t is the spatial pattern at time t and Y_t is a vector of variables that affect the transition [Sklar and Constanza 1991]. The raster data structure, commonly used in GIS is the model of space underlining most spatial simulations. The simulation defines the flows or processes that occur among cells. The most common way to model these spatial processes is ordinary and partial differential equations and cellular automata.

A common approach to study these models is to capture the dynamic using Simulation while a Geographic Information Systems (GIS) software is used to visualize and analyze the spatial distribution of the model variables. The simulation program is written in a general programming

language to solve a particular type of problem and results are exported periodically to a GIS software. Programs are written to translate the information supplied by the simulation to GIS and viceversa. This “loose coupling” approach [Neyrges, 1993] requires expert users that are not only familiar with the particular environmental problem that is being modeled but also with the programming language and the GIS software that is being used. This complicates the study and increases considerably the development times of a project.

Another approach is to incorporate, in the same interface, the simulation program and the main functions of GIS (e. g. AirWare [Environmental Software & Services, <http://www.ess.co.at>]). A friendlier and unique user interface is provided which eases considerably the development of the project. This, however, comes with a price: the tools are designed only for a very specific kind of problem and the user is not allowed to directly manipulate the basic equations and assumptions of the model.

This paper presents *SpaSim*, a software designed to overcome the main disadvantages of the foretold approaches. *SpaSim* allows the specification, simulation, visualization and analysis of spatial models in the same environment, using a friendly user interface while at the same time providing considerable flexibility. Although the software was designed to run spatial simulation models in general, currently only cellular automata models have been implemented. In what follows, a general description of the software is presented. To further clarify the use of *SpaSim*, an example simulation is provided. The paper concludes highlighting the main features of the software.

2. SOFTWARE DESCRIPTION

SpaSim was initially inspired and conceived as a library of software components to support the development of spatial models in GLIDER [Domingo et al. 1996] and its current development environment, GALATEA [Dávila and Uzcátegui 2000]. Current implementation of *SpaSim* allows model specification using cellular automata. Cellular automata is increasingly being used as a method to study the spatial and temporal of environmental systems [e.g. Karafyllidis et al. 1997; Dunkerly 1999].

The library was designed using the unified modeling language (UML) [Muller, 1997] and developed using Java. It has 5 packages:

- *SpatiaModel* contains the classes to model a raster data model: *RasterMatriz* that implements the raster model as a matrix and *RasterQuadtree*, the implementation using quad trees.
- *SpatialTemporalModel* includes the object classes that implement two of the data models most used in temporal GIS: the snapshot model (*RasterSnapshot*) and the space-time composite model (*RasterSpatioTemporalComp*) [Peuquet 1998]. The snapshot model is probably the most common approach to store dynamics in a GIS: the state of the simulation in a given time is stored and a time tag is associated with the layer. The model is flexible and allows saving the state of the model in different instants or periods of time but is inefficient in terms of data storage; the specific instant where a change occurs cannot be determined and changes can be explicitly recovered only comparing cell to cell in contiguous states. On the other hand, in the space-time composite model given an initial state represented by a raster layer a detailed account of the modifications are registered in cell by cell basis. The model avoids data redundancy and stores all the changes but complicates the reconstruction of a state layer in a given time.
In a simulation both models are needed and complement each other. The state of the model may be needed in a particular time or set of times or it may be necessary to reconstruct completely the evolution or history of the system in a given period or time. Thus, both models are included in our software.
- *SimulationSpatialModel* consists of all the object classes that implement the cellular automata model, in particular the classes *Neighborhood* and *Rules* that define the neighborhood and rules that characterize the cellular automata.
- *SimulationSpatial* comprises the object classes that implement the simulation engine for all the models defined in the package *SimulationSpatialModel*
- *Graphics* contains all the necessary classes to allow visualization of raster maps.

SpaSim has a graphical use interface that integrates in a manner transparent to the user all the components previously described.

There are several tools to perform simulation using cellular automata. Some examples are CASE [Burgess et al., 1996], CASim [Freiwald and Weimar, 1999] and CAM [Belso and Vargyas 1993]. Together with these tools, our software

allows the simulation and visualization of models based on cellular automata. Unique to *SpaSim* is the structures to save the spatial evolution of automata which offer the capacity to perform spatial and spatio-temporal analysis.

The software consists of four modules: model, simulation, visualization and analysis. Model definition or retrieval of a previously stored model should be done previously to any other operation.

2.1 Model Definition

The specification of the cellular automata is done in terms of the definition of the grid or model space, neighborhood and rules. Two different options are offered to store the changes of model through time: based on the time composite model (defined as *Cellular History* in the software) and based on the snapshot model (defined as *Cellular Snapshot*). The automata may have different layers and a different type of neighborhood can be defined for each layer, including multidimensional neighborhoods. Neighborhoods are specified using a triplet (x,y,z) to denote a neighbor located at position (x,y) in the plane and in the automata layer z . The coordinate system is relative to the cell that is currently being processed which has the coordinates $(0,0,0)$. For example, the set: $(-1,-1,0)$ $(1,-1,0)$ $(-1,1,0)$ $(1,1,0)$ defines the neighborhood shown in figure 1:

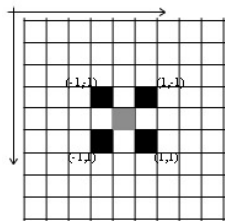


Figure 1: Neighborhood $(-1,-1,0)$ $(1,-1,0)$ $(-1,1,0)$ $(1,1,0)$

Rules are specified following the convention: *If <condition> then <action>*. Both, condition and action, can be simple or composite. Simple conditions are expressed as $f(variables)$ *op valor* where *op* is one of the logical mathematical operators and f is one of the predefined functions in table 1 and predefined actions consists of actualizing or incrementing the value of a given cell.

Composite conditions relate simple conditions with logical operations such as *and*, *or* and *not*. A detailed account of the predefined functions for conditions and actions is given in [Moreno 2001].

Function	Description
Act (layer)	Returns the current cell value in layer
cell (layer, x, y, z)	Returns the value of the neighbor cell in the given location
Sum (layer)	Returns the sum of all the neighbors of the current cell in layer
Vecinosedo (layer, value)	Returns the number of neighbor of the current cell whose value is equal to value

Table 1: Predefined functions

2.2. Simulation

In this module the simulation is run. Specification of the simulation time and the time step of the simulation are required.

2.3. Analysis

Two types of analysis operations are provided: Spatial and Spatio-temporal. Spatial analysis operations include reclassification, scalar operations and overlay. The dialogs to perform these functions are straightforward and very similar to the ones provided in popular GIS software like IDRISI (<http://www.clarklabs.org/>).

The spatio-temporal analysis operations allow answering questions that frequently arise when looking at output of a simulation. For example, the modeler may be interested in knowing where and when was achieved a given model metric (minimum, maximum, etc) and what was its value; or he or she may require the map of a given model metric for a period or all of the simulation time. The answer to first kind of question is a collection of values indicating time, position and magnitude. On the other hand, the answer to the second kind of question is a map. We call the first type of inquiry theme or value queries and the second type spatial or map queries.

2.4. Visualization

Model layers are shown using the Visualization module. The visualization window dialog is dynamic and contains several tabs, one for each of the automata layers. For each layer the user selects the time and color pallet to be used. Also GIS layers can be exported, imported or saved as independent layers. In the next section, the use of *SpaSim* is explained with a model of epidemic propagation.

3. EXAMPLE: A MODEL OF EPIDEMIC PROPAGATION

The model presented is a simplified version of a model of epidemic propagation due to [Sirakoulis et al. 2000] and describes the effects of vaccination and the movement of a population on the propagation of an epidemic. Each cell in the automata represents a portion of the total population that can be in any of the states: infected, immune and susceptible. An infected cell remains infected a total of t_{im} time units. After that, the cell remains immune a total time of t_{im} time units. Following the end of the period of immunity, the cell becomes susceptible again.

3.1 Model definition

Figure 2 presents the initial steps to create a new model. It is necessary to specify the type of spatial model, Cellular Automata, and the structure to save simulation results.

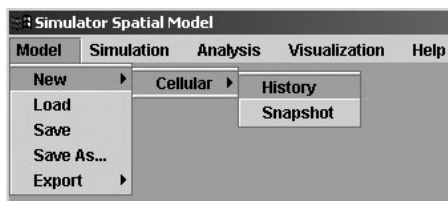


Figure 2: SpaSim initial window.

The next step consists in defining the automata. The program asks for the number of layers and its dimensions. After pressing the button *Generate*, a new window panel appears with n tabs, one for each layer. The model of epidemic propagation has 3 layers: infectious state, called *Epidemics*; time infected, called *TimeInf* and immune time, called *TimeInm*. Figure 3 shows the definition of layer 0, the infectious state that indicates if a cell has been infected (denoted by 0), is immune (denoted by 1) or susceptible (denoted by 2). The name of the layer, its minimum and maximum state values, cell resolution and no data flag need to be specified.

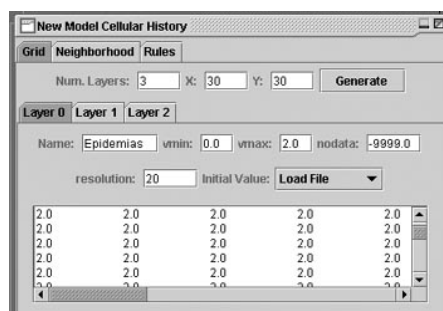


Figure 3: Layer definition

Layer 0 is initialized with all cells in 2 (susceptible) but two cells. There are two infected cells. One in row 10, column 20 and another one in row 20, column 10. The layers corresponding to infected time and immune time are defined in a similar manner as layer 1 and layer 2. They are initialized with all the values in 0.

After specifying the layers, the neighborhoods of each layer are defined selecting the tab *Neighborhood*. All the neighborhoods in this example are of Moore type, one already predefined.

In order to demonstrate the writing of the rules, let $E_i(t)$ be the state that represents the infectious state (layer 0) in cell i at time t ; $Tinf_i(t)$ the elapsed time since infection of cell i (layer 1) and $Tinm_i(t)$ the immune time of cell i at time t .

The first rule of the model is that if cell i is infected and the time elapsed since infection is less than t_{im} (taken to be 5 in this example) it is necessary to increase the time elapsed since infection, i.e., if $E_i(t) = 0$ and $Tinf_i(t) < t_{im}$ then $Tinf_i(t+1) = Tinf_i(t)+1$

In *SpaSim* this rule is written as: `act(0)=0 and act(1)<5 then incell(1,0,0,1)`. Rules can be written directly in the space provided or using the menu lists as shown in figure 4.

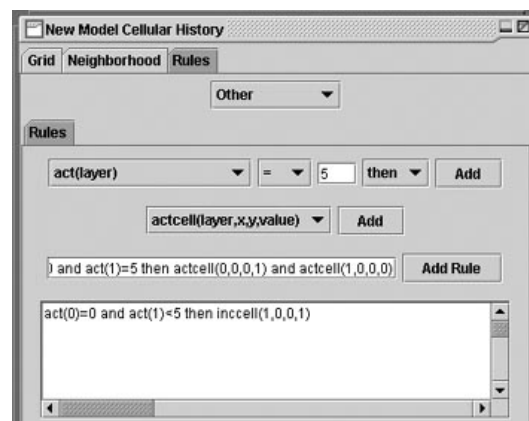


Figure 4: Rule construction window

Table 2 shows all the rules used in the model and its writing in *SpaSim*.

Once the model description is completed the simulation is run for 20 time units. Model results can be visualized at any time. For example, figure 5 shows layer 0 with the initial values and after 8 unit simulation times.

Rule	<i>SpaSim</i>
If $E_i(t) = 0$ and $Tinf_i(t) < t_{im}$ then $Tinf_i(t+1) = Tinf_i(t)+1$	if act(0)=0 and act(1)<5 then inccell(1,0,0,1)
if $E_i(t) = 0$ and $Tinf_i(t) = t_{im}$ then $E_i(t+1) = 1$ and $Tinf_i(t+1) = 0$	if act(0)=0 and act(1)=5 then actcell(0,0,0,1) and actcell(1,0,0,0)
if $E_i(t) = 1$ and $Tinm_i(t) < t_{im}$ then $Tinm_i(t+1) = Tinm_i(t)+1$	if act(0)=1 and act(2)<10 then inccell(2,0,0,1)
if $E_i(t) = 1$ and $Tinf_i(t) = t_{im}$ then $E_i(t+1) = 2$ and $Tinm_i(t+1) = 0$	if act(0)=1 and act(2)=10 then actcell(0,0,0,2) and actcell(2,0,0,0)
if $E_i(t) = 1$ and one or more neighbors are infected then $E_i(t+1) = 0$	if act(0)=2 and vecinosedo(0,0)>=1 then actcell(0,0,0,1)

Table 2: Epidemic propagation rules on *SpaSim*

Finally, queries about different aspects of simulation results can be performed using the analysis module. For example, consider the following question “What was the total area infected in the square region delimited by cell (4,4) as the upper left corner and cell (15,15) as the lower left corner at time 10?” This is a theme or value query. The question is formulated using the menu list provided as shown in figure 6. The answer to the question is 1360 as given in the results panel.

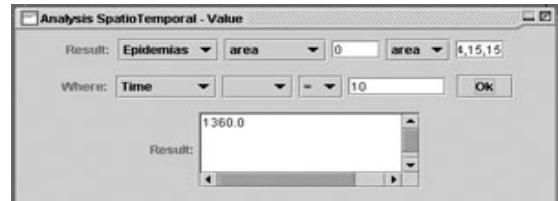


Figure 6: Formulation of a theme or value query

Other types of queries involve the spatial distribution of cells with a particular characteristic. For example, it may be interesting to show a map of infected cells over the first ten units of time. Figure 7 shows the formulation of the query in *SpaSim* and figure 8 shows its results.

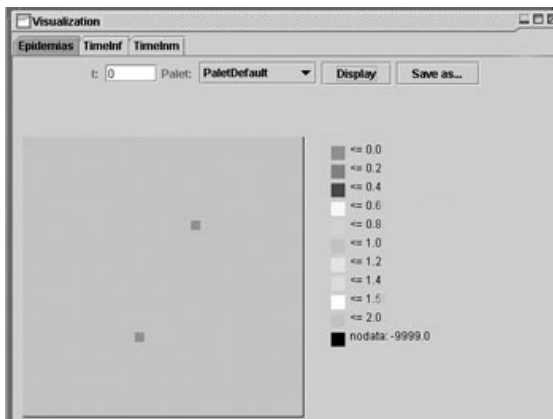


Figure 5-a: Initial model conditions for infectious state (layer 0).

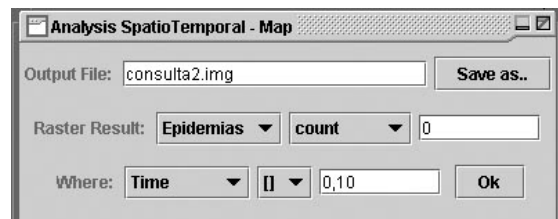


Figure 7: Formulation of a map query

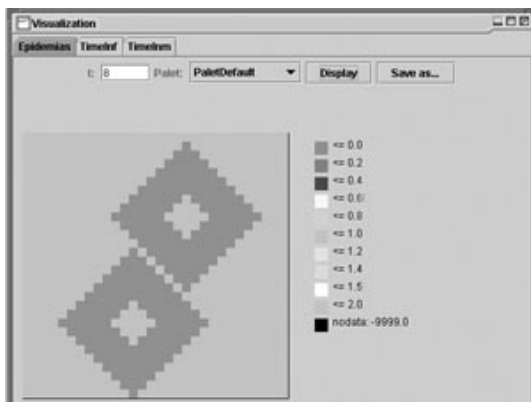


Figure 5-b: Model results for infectious state (layer 0) at time 8

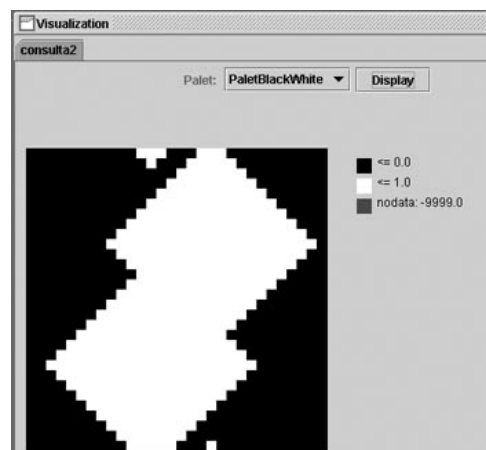


Figure 8: Map showing the distribution of infected cells over the first 10 time units of simulation.

4. CONCLUSIONS

SpaSim is a software for running spatial simulations that consolidates in one application the resources needed to define, simulate, visualize and analyze spatial models based on cellular automata. The software includes modules to:

1. Define spatial models in terms of cellular automata. The models may have multiple layers and different neighborhoods can be defined in each layer.
2. Simulate the models during a specified period of time and save the results of simulation, using one of the two data space-temporal data structures provided.
3. Visualize the different layers of the automata in a given time or any other raster layer exported into the system.
4. Perform spatial and spatio-temporal queries of simulation results.
5. Export the cellular automata model to java source code so that extensions or modifications not allowed directly in *SpaSim* can be performed by the user.

Further development of *SpaSim* will include spatial models solved using ordinary and partial differential equations and refinements of the rule object class to encompass a greater number of rule types.

5. ACKNOWLEDGMENTS

This research was supported by a grant awarded by el Consejo de Desarrollo Científico, Humanístico y Tecnológico (CDCHT) of Universidad de los Andes (I-524-95-02-AA)

6. REFERENCES

- Acevedo, M.F.; D.L. Urban and M. Abla, Transition and Gap Models of Forest Dynamics, *Ecological Applications*, 5(4): 1040-1055. 1995
- Belso, Z. and Vargyas, M. Department of Computer Science, Eotvos University, Hungary. (<http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/ca/systems/cam/0.html>), 1993.
- Burgess, M., J.A. Mikkelsen. and T.E. Sevaldrud, CASE: Cellular Automata Simulation Environment. Centre of Science and Technology, Oslo College (http://www.iu.hio.no/~cell/CASE_toc.html).1996
- Dávila, J. and M. Uzcátegui,. GALATEA: A multi-agent simulation platform. Proceedings of MSNN'2000: International Conference on Modeling, Simulation and Neural Networks, 2000.
- Domingo C., M. Sananes and G. Tonella, GLIDER Reference Manual, IEAC-CESIMO, Universidad de los Andes, Mérida, Venezuela. 1996.
- Dunkerley D. L., Banded chenopod shrublands of arid Australia: modelling responses to interannual rainfall variability with cellular automata. *Ecological Modelling*, 121(2-3).1999
- Freiwald, U. and J. Weimar. CASim: JAVA Environment for Simulating Cellular Automata. (<http://www.sc.cs.tu-bs.de/~weimar/jcasim/casimhand/handbook.html>). 1999.
- Gardner, R.H; Hargrove, W.W.; Turner, M.G.; Romme W.H and D.G. Despain. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling*, 135 (2-3) 2000.
- Karafyllidis I. and A. Thanailakis, A model for predicting forest fire spread using cellular automata. *Ecological Modelling* 99(1).1997
- Moreno N. *Diseño e Implementación de una estructura para el soporte de simulación espacial en GLIDER*. Tesis de Maestría, Postgrado de Computación, Universidad de los Andes, Mérida, Venezuela. 2001
- Muller, P. *Modelado de objetos con UML*. Eyrrolles. Ediciones Gestión 2000, S.A. Barcelona, España. 1997.
- Neyrges, T.L. Understanging the scope of GIS: its relationship to environmental modeling. In: Goodchild, M.F, B.O Parks and L.T. Steyaert (editors), *Environmental Modeling with GIS*. Oxford University Press, New York, 1993.
- Peuquet, D. Time in GIS and Geographical Databases. In: Longley et al (editors) *Geographic Information Systems: Principles, techniques, applications and management*,1998.
- Sirakoulis, G. C., I. Karafyllidis. and A. Thanailakis. A cellular automaton model for the effects of population movement and vaccination on epidemic propagation. *Ecological Modeling* 133(3) pp 209-223. 2000
- Sklar F. H; Constanza R. Dynamic Spatial Models, In: Monica G. Turner y R. H. Gardner (editors), *Quantitative Methods in Landscape Ecology*, Springer, 1991