Vectorial data use in a m:n-AC^k cellular automaton.

Pau Fonseca i Casas[1]

[1]Barcelona School of informatics computing laboratory, Campus nord, Ed B6, Jordi Girona 1-3,
08034 Barcelona Spain
Tel. (+34)934017732 Fax (+34)934017040
pau@fib.upc.edu

KEYWORDS: Cellular automaton, vectorial layer, m:n-AC^k, GIS.

1. Introduction

The purpose of this paper is present the m:n-AC^k cellular automaton, a generalization of the classical cellular automaton, allowing the use of GIS vectorial data in a simulation model. Also m:n-AC^k allows working with different layers in a single cellular automaton.

The paper first describes the classical cellular automaton structure and the m:n-AC^k structure, and secondly shows an example of propagation using vectorial data obtained directly from a GIS and remarks the importance of the topological space used to represent the automaton space.

2. Cellular automata

Cellular automata are discrete dynamical systems whose behaviour is completely specified in terms of a local relation (Emmeche C., 1998). Cells represent automaton space; time advances in discrete steps following “the rules”, the laws of “automaton universe”, usually expressed in a small look-up table. At each step every cell computes its new state in function of its closer neighbours. Thus, system's laws are local and uniform.

Next figure shows one-dimensional cellular automaton initial state and successive two states after rules application.

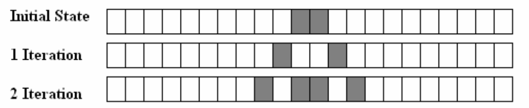

Figure 1:One-dimensional cellular automaton

3. MULTI:N-DIMENSIONAL CELLULAR AUTOMATON (M:N-AC)

A multi:n-dimensional cellular automaton (m:n-AC) is a generalization of a cellular automata defined as follows (Fonseca et al. 2005):

Definition 1: m:n-AC^k

A multi n dimensional cellular automaton is a cellular automaton generalization composed by m layers with n dimensions each one.

The representation is:

$$m{:}n - AC^k$$

Where

m: is the automaton number of layers.

n: is the different layers dimension.

k: is the number of main layers (1 by default). A layer in a m:n-AC^k is a main layer if a transition function Λ is defined in order to modify its state. A m:n-AC automaton only presents one main layer, while m:n-AC^k automaton presents k main layers.

A two dimensional cellular automaton is represented by a 1:2-AC. A transition in a m:2-AC cellular automata is defined as in a 2-dimensional cellular automata, but main layer cell state is a combination of data contained in the m-1 secondary layers at the same position.

All layers must be georeferenced. The GIS data classification is shown in the next table based in the table of (Fonseca et al. 2004).

Layer	GPS integration	Description
2DLayers	Geo referenced	Point, polylines, texts or lines.
3DLayers	Geo referenced	Fixed population of elements over a matrix, and DEM.
Routes	Track points	Represent *Objects* movement.
2DObjects	Waypoints	a 2D object in an specific position
3DObjects	Waypoints	a 3D object in an specific position.

Table 1. GIS data classification in a simulation model.

Suitable data that can be represented in the m:n-AC^k layers are vectorial data (2DLayers) or raster data (3DLayers). Other elements can be represented using common simulator elements.
Since multiple layers belong to a single automaton its state is defined as follows.

Definition 2: $E_m[x_1,..,x_n]$, layer m state in $x_1,..,x_n$ position
E_m is a function describing cell state in position $x_1,..,x_n$ of layer m.

E_m function allows state representation for each cell in the different layers of the automata, but this is not the global state of the automata. This state is represented by the *EG* function.

Definition 3: $EG[x_1,..,x_n]$, automata status in $x_1,..,x_n$ position.
EG returns automata global state in position georeferenced by coordinates $x_1,..,x_n$.

The global state of cellular automata depends on EG function in all automata positions.
Combination functions Ψ is represented by equation:

$$\Psi(E_1[x_1..x_n];^{m-2)},E_m[x_1..x_n])= EG[x_1..x_n] \qquad (1)$$

In a common cellular automaton, evolution function allows global automata state change through cells value modification.
In a m:n-AC^k vectorial layers use makes necessary to generalize the neighbourhood and later define a new function that determines something similar to cell size (nucleus function).

Definition 4: Evolution Function Λ_m
Function defined for the layer m to modify its state through the state of others layers using combination function Ψ, and vicinity and nucleus functions.

Intuitively evolution function allows the representation of the modifications in this layer (modifications in nucleus area of a point $x_1.x_n$), using the state of other layers with combination function Ψ, and the vicinity area.

4. Neighbourhood definition extension, vicinity and nucleus concept.

In a traditional cellular automaton neighbourhood function must be defined in order to determine the cells to be considered in the evolution function.
In a m:n-AC, due to the vectorial representation capability, is necessary to redefine the concept without using cells with the help of some kind of georeference.
Therefore is necessary to define the space area characterizing neighbourhood without cell dependency. **Vicinity function** defines, from a position $x_1,..,x_n$, the points to be considered inside evolution function in new layer state calculus. **Nucleus function** allows to define, from a position $x_1,..,x_n$, the environment to be modified after evolution function calculation. Neighbourhood concept is related to topological concept formalizing a colloquial concept. Remembering topology mathematical definition, a topological space is a non empty set X with a defined topology. Representation is (X, T). If (X, T) is a topological space and p a point of X, a subset “A” of X is a

neighbourhood of p if and open U of the topology T exist as $p \in U \subset A$.
Existing relation between mathematical topology and vicinity and nucleus concepts allows formalizing the points order in layers in two levels:

1. First level represents points to be considered in order to calculate new state.
2. Second level represents points to be modified once state changes.

Mathematical topological concept permits explicit definition of neighbourhoods for different points. Hence for a raster layer (discrete space) neighbourhood for each point can be explicitly defined.

Definition 5: vicinity topology
Topology defining the set of points (neighbourhood) for layer m, to be considered for Λ_m calculus.

In a similar way for nucleus topology:

Definition 6: nucleus topology
Topology defining the set of points (neighbourhood) for layer m, to be modified by Λ_m calculus.

These two topologies define neighbourhood structures necessary for each point in order to establish vicinity and nucleus. In despite of that, not all neighbourhoods can be used to represent nucleus or vicinity, and only one set can be used.
To define the set to be used from a point neighbourhood, usually can be necessary to define a metric, based for instance in Euclidian distance:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{3}$$

Distance d(x,y) allows the definition of neighbourhood bases as:

$$B(x,r) = \left\{ y \in \Re^m \,/\, d(x,y) < r \right\} \tag{4}$$

This is the usual topology on RxR (Arregui Fernandez J., 1988), can be one of the more indicated topologies for a m:2-AC based in the RxR space defined with the usual distance.
In a general way we can define a distance r from the point x defining **restrictions** of the selected neighbourhood.
A typical restriction rule can be defined, for instance, as calculate minimum neighbourhood containing all the points accomplish $d(x,p)<r$. In the usual topology presented in (4), B(x,r) are the minimum neighbourhood to accomplishes this restriction.
In an m:n-AC two restrictions rules must be defined, one for the vicinity topology, and other for nucleus topology. These two restriction rules are used to construct vicinity and nucleus functions.
Now we can define vicinity and nucleus functions.

Definition 7: Vicinity function vn(x_1,..,x_n)
Function returning minimum open set of vicinity topology containing point $x_1,..,x_n$, and including maximum points that accomplishes the restriction and minimum points not accomplishing the restriction.

If the restriction is defined through usual distance its represents neighbourhood containing maximum points that accomplish $d(t,p)<r$ and minimum points that accomplish $d(t,p)\geq r$.
In the same way we define the nucleus function.

Definition 8: Nucleus function nc(x_1,..,x_n)
Function returning minimum open set of nucleus topology containing point $x_1,..,x_n$, and including maximum points that accomplishes the restriction and minimum points not accomplishing the restriction.

5. Example: 1:2-AC on R^2.

In this example GIS data in vectorial format must be used to evaluate a propagation of “something”. The data, represented in IDRISI format are shown in the next lines. Pixels are the unit used.

```
file title  : Initial surface
id type     : integer
file type   : ascii
object type : polygon
ref. system : pixel
ref. units  : pixel
unit dist.  : 1
min. X      : 0
max. X      : 2160
min. Y      : 0
max. Y      : 2580
pos'n error : unknown
resolution  : unknown
```

Figure 2: dvc IDRISI vectorial file.

```
67.0000000 4.0000000
300 200
300 250
400 300
500 200
0.0000000 0.0000000
```

Figure 3: vec IDRISI vectorial file.

First is necessary to define two topologies (vicinity and nucleus). These two topologies can be defined by the equation (4) with an r specified for vicinity topology and an $r' \leq r$ for the nucleus topology. Restrictions, that determines vicinity and nucleus functions, can be defined by returning a set that contains maximum number of points accomplishing $d(x,p)<r$ and minimum number of points with $d(x,p) \geq r$ (r’ for the nucleus topology).

Hence the vicinity and nucleus functions are:

- $vn(x_1,..,x_n)$=returns the open set centred in the point x_1,x_2 for the topology that defines the vicinity.
- $nc(x_1,..,x_n)$=returns the open set centred in the point x_1,x_2 for the topology that defines the nucleus.
- r =10, and r'=5 (pixels are the unit).

Implementation of Λ_m over Z^n since it corresponds to the usual way of using a cellular automaton. Data initial representation is shown in the next figure.

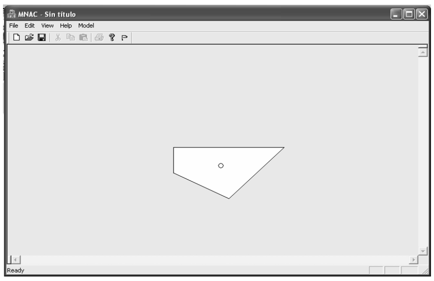

Figure 4: Polygon initial state.

We wish to simulate the propagation of this polygon over the area. We first define state function. The automata is an 1:2-AC EG=E_1

$E_1(x_{i,j})$ =

- Filled: "f", if $x_{i,j}$ belongs to polygon.
- Empty: "o", if $x_{i,j}$ doesn't belong to polygon.

In this automata time step is discrete and space is continuous (R^2).

Evolution function is defined follows:

$\Lambda_1(x_{i,j})$ = "o" to "f" if $E_1(x_{i,j})$="f" and exists $x_{i',j'} \in$ vn($x_{i,j}$) / E($x_{i',j'}$)="o"

In the next figure the shape after 11 steps is shown.

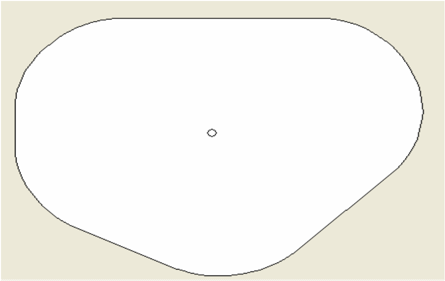

Figure 5: vectorial evolution.

The area after 11 applications of evolution function is **104544.6.** Some complexity exists related to evolution calculation on R^n this can be reviewed at (Pau Fonseca et alt. 2005).

Using a common cellular automaton, in fact a 1:2-AC over Z^2, the estate definition E_1, and the evolution function are the same, but due to the vicinity and nucleus topology works in Z^3, the *vn* and *nc* functions must be redefined.

The nucleus topology defines, for each cell, one set that are composed by the cells surrounding the cell defined by the coordinates x_i, x_j. with a radius of 5 pixels. For the vicinity topology the radius is 10 pixels.

Following these conditions, the distance in this space is the only element that must be redefined.

If the distance is the distance defined using (3), *vn* and *nc* functions can be represented in the next figure (the dark cells belongs to *vn*, while the light grey cells belongs to *nc*)

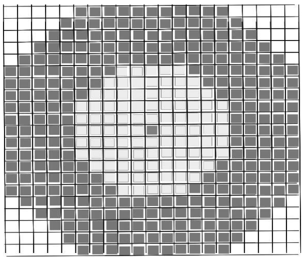

Figure 6: raster *vn* and *nc* functions using (3).

Therefore, distance can be defined depending on the number of cells. In that case the *vn* and *nc* functions can be represented like is shown in the next figure:

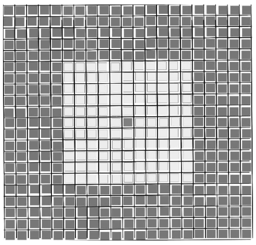

Figure 7: raster *vn* and *nc* functions using cell number.

With the second example over Z^2 surface final aspect is shown in the next figure.

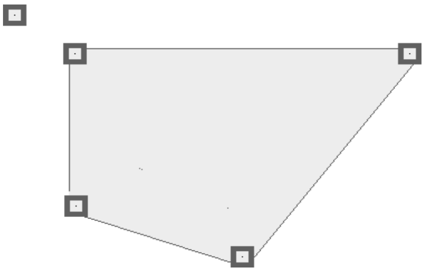

Figure 8: raster evolution.

7. CONCLUSIONS

The m:n-CA^k cellular automaton is presented, a generalization of classical cellular automaton without dependency on implementation used to represent it.

This extension of cellular automaton allows vectorial GIS data use inside a discrete simulation model and the use of different layers in a single cellular automaton. This simplifies GIS data use in a complex simulation model due to all the GIS data can be represented in a single structure.

A comparison for the same evolution function is presented using different topological spaces, remarking the independence of this new theoretical approach with the implementation method used, and the importance of the topological space used for the results accuracy.

References

Arregui Fernandez J., 1988, *Topología* Ed. UNED

Benenson I. and M.Torrens P. 2004 *Geosimulation: Automata-based modelling of urban phenomena* Ed Wiley

Emmeche C.1998. *Vida Simulada en el ordenador* Ed. Gedisa.

Fonseca P.; Casanovas J.; Montero J. 2003. *LeanSim: Un sistema de simulación para el entrenamiento de personal especializado dentro de sistemas complejos*. Memorias de la 2ª Conferencia Iberoamericana en sistemas, cibernética e informática CISCI 2003, Volumen I. 318-323

Fonseca P.; Casanovas J., Montero J., 2004. *GIS and simulation system integration in a virtual reality environment.* Proceedings GISRUK 2004, 403-408

Fonseca P.; Casanovas J., Montero J 2004. *A cellular automata and intelligent agents use to model natural disasters with discrete simulation.* Proceedings EMS 2004.

Fonseca P.; Casanovas J., Montero J 2004. *Leansim® virtual reality distributed simulation suit.*; Proceedings MSO 2004.

Fonseca P.; Casanovas J. 2005. ESS205, *Simplifying GIS data use inside discrete event simulation model through m_n-AC cellular automaton*; Proceedings ESS 2005.

Geoffrey Gordon. System simulation; Editorial Prentice Hall 1978

Guasch A.; Piera M.A.; Casanovas J.; Figueres J. 2002. "Modelado y simulación. Aplicación a procesos logísticos de fabricación y servicios." Ediciones UPC.

Law, A. M., Kelton, W. D. 2000. *Simulation modelling and analysis.* McGraw-Hill.

Robinson, A. and others, 1987. *Elementos de Cartografía*. Omega, Barcelona.

Wainer, G. and Giambiasi, N. 2001. *Timed Cell-DEVS: modelling and simulation of cell spaces. Discrete Event Modeling & Simulation: Enabling Future Technologies.*, Springer-Verlag. 2001

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

Biography

PAU FONSECA obtained his degree in computer engineer on 1999 from Technical University of Catalonia. Now is full time professor of this university, teaching in statistics and simulation area. Also he works in the LCFIB (Barcelona informatics school computing laboratory) developing Simulation projects since 1998, and is member of LogiSim group dedicated to the research and simulation tools development.