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

GIS data classification used is shown in the next table:

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.

In a simulation environment suitable data that can be represented in the m:n-AC^k layers are vectorial data (2DLayers) or raster data (3DLayers). Other elements (Routes, 2DObjects or 3DObjects) can be represented using common simulator elements. A two dimensional cellular automaton is represented by a 1:2-AC.

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 figures.

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

IDRISI VEC file

```
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
```

IDRISI DVC file

First is necessary to characterize two topologies (vicinity and nucleus), defining the Euclidean distance with an r specified for vicinity topology and an r'≤r for the nucleus topology. 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 L_m over Z^n since it corresponds to the usual way of using a cellular automaton.
$E_1(x_{i,j})$ =

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

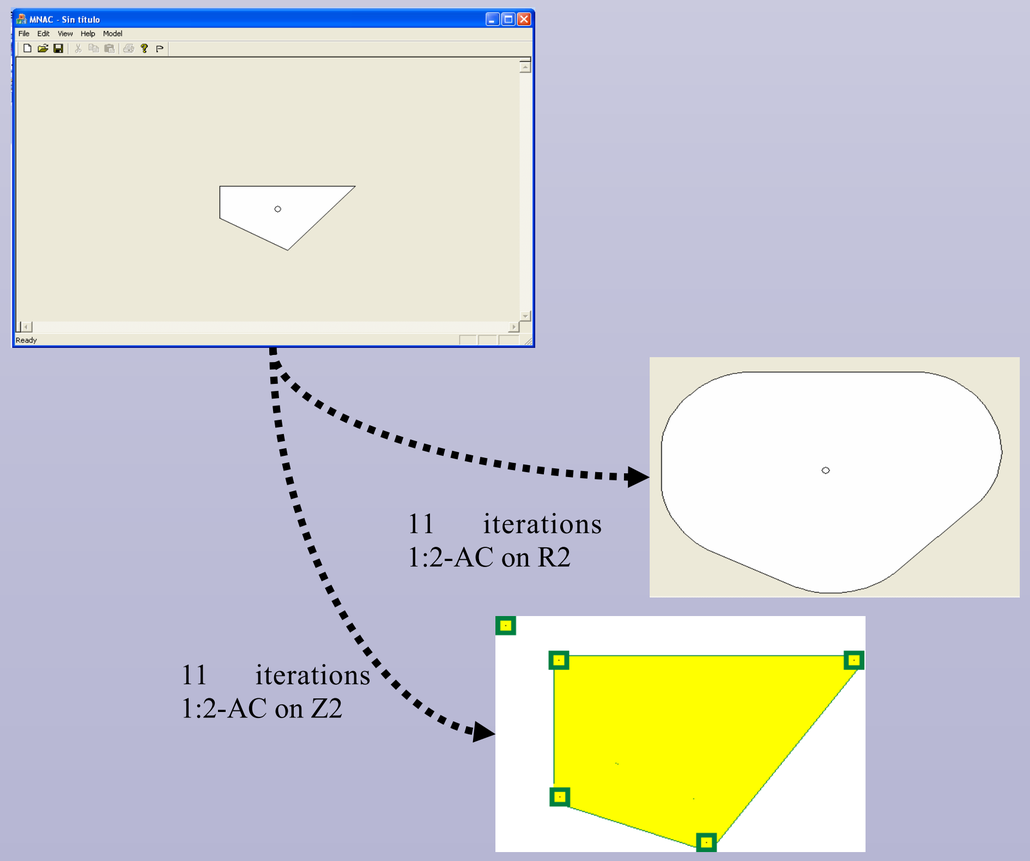
In this automata time step is discrete and space is continuous (R^2).
Evolution function is defined follows:

- $\Lambda_1(x_{i,j})$ = "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 figure the shape after 11 steps is shown.

Example: 1:2-AC on Z^2.

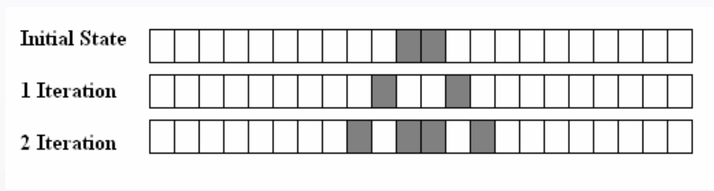
State definition E_1, and evolution function are the same that over R^2, but due to the vicinity and nucleus topology works in Z^2, vn and nc functions must be redefined.
The nucleus topology defines, for each cell, one set that are composed by the cells surrounding the cell defined by the coordinates xi, xj. with a radius of 5 pixels. For the vicinity topology the radius is 10 pixels.
The distance used is $d(x, y) = \max(abs(x_1 - y_1), abs(x_2 - y_2))$



Conclusions

The m:n-CAk 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. The two evolutions helps to remark first the independence of this new theoretical approach with the implementation method used and second, the importance of the topological space used for the results accuracy.

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



Theory: m:n-AC^k

A multi n dimensional cellular automaton is a cellular automaton generalization composed by m layers of dimension n.

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

Where

- m: is the automaton number of layers.
- n: is the 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.

Em is a function describing cell state in position x1,..,xn for layer m.

$\Psi(E_1[x_1..x_n],\cdot^{m-2)},E_m[x_1..x_n]) = EG[x_1..x_n]$ EG returns automata global state in position referenced by coordinates x1,..,xn.

The global state of cellular automata depends on EG function in all automata positions. Combination functions Ψ allows the combination for the data that belongs to different cellular automaton layers.

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

Evolution Function Λm
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.

Vicinity topology
Topology defining the set of points (neighbourhood) for layer m, to be considered for Λm calculus.

Vicinity function vn(x1,..,xn)
Function returning minimum open set of vicinity topology containing point x1,..,xn, and including maximum points that accomplishes "the restriction" and minimum points not accomplishing "the restriction".

Nucleus topology
Topology defining the set of points (neighbourhood) for layer m, to be modified by Λm calculus.

Nucleus function nc(x1,..,xn)
Function returning minimum open set of nucleus topology containing point x1,..,xn, and including maximum points that accomplishes "the restriction" and minimum points not accomplishing "the restriction".

Vicinity function defines, from a position x1,..,xn, the points to be considered inside evolution function in new layer state calculus. Nucleus function allows to define, from a position x1,..,xn, the environment to be modified after evolution function calculation.

To characterize vicinity and nucleus functions usually a metric must be defined, for instance the Euclidian distance:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (1)$$

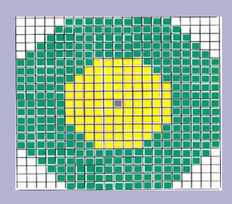
Or

$$d(x, y) = \max(abs(x_1 - y_1), abs(x_2 - y_2)) \quad (2)$$

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

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

Restriction example: A typical vicinity and nucleus function can be defined for a point p for instance, as the set containing maximum points that accomplish d(t,p)<r and minimum points that accomplish d(t,p)≥r .
The representations for the nucleus and vicinity functions for a point p over R^2 depends on the distance selected (the green cells belongs to vn, while the yellow cells belongs to nc)

With (1)  or with (2)