

# An Architecture for Agent-based Modelling and Simulation of Geospatial Phenomena

Pedro Ribeiro de Andrade<sup>1</sup>, Antonio Miguel Vieira Monteiro<sup>1</sup>,  
Gilberto Camara<sup>1</sup>, Tiago Garcia de Senna Carneiro<sup>2</sup>

<sup>1</sup>INPE – National Institute for Space Research

<sup>2</sup>UFOP – Federal University of Ouro Preto

{pedro, miguel, gilberto}@dpi.inpe.br, tiago.carneiro@iceb.ufop.br

***Abstract.** This work presents a general architecture for building and simulating agent-based models that use real-world geospatial data, take into account all the ways geospatial data can feed these models. We focus on how data can be used to create an initial arrangement for the model, as if it was a static representation. We have as hypothesis that the Generalized Proximity Matrix (GPM) is a foundation for setting up the relations between the entities of an agent-based model for simulating geospatial phenomena.*

## 1. Introduction

Representing the micro-behaviour in order to get a macro-pattern is the key challenge of agent-based modelling (ABM). Since its beginning, real world evidences guided the development of agents and the environment with which they interact. This has led to the development of artificial worlds to understand underlying principles of human behaviour, such as wealth distribution and evolution of cooperation (Axelrod 1997; Epstein and Axtell 1996).

On the other hand, many models have to be less general and more precise, mainly when we study behaviour on the geographic space, as in the case of land use and land cover change. In these models, the agents have to be grounded to a representation of the world, moving from theoretical abstractions to case-based models that focus on populations within specific localities. With it, agent based modelling has facing the challenge of using not only real world evidences but data, especially geospatial data (Gilbert 2008), to feed models. Brown and others have argued that “incorporating real spatial heterogeneity into agent-based models will improve our ability to draw conclusions about the behavior of complex systems in realistic environments” (Brown et al. 2005). This leads to a need towards integrating toolkits for agent-based modelling and Geographic Information Systems (GIS), since the formers need the spatial heterogeneity and the latters are not well suited for representing and studying behaviour (Goodchild 2005).

However, most of the available toolkits were initially conceived to work with landscapes that do not consider the environment in which we move and interact. They rely on non-grounded agents, in the sense that they do not represent entities which can sense and act in a representation of the real world. It makes implementing models based on the geographic space a difficult task, once the modeller has to carry out all the necessary geospatial data input and output.

To follow the current development, toolkits have to carry out the support for working with geospatial data. However, the demands and formalisms for creating agent-based models at this complexity level are not well established. Newest toolkits, as well as spatial extensions of older toolkits, have been created according to the class of problems they are trying to support, instead of following a conceptual and more general approach. Common approaches represent landscapes where artificial agents move and interact using geospatial data (Barros 2004; Crooks 2007), or use data to feed agents that move in artificial environments (Crooks 2007), but they do not use geospatial data to connect them.

The objective of this work is to present an architecture to allow implementing and simulating agent-based models, taking into account all the ways geospatial data can feed these models. We focus on how data can be used to create an initial arrangement for the model, as if it was a static representation. Our working hypothesis is that the idea of Generalized Proximity Matrix, or GPM, is a foundation for setting up relations between the entities of agent-based models that simulate geospatial phenomena.

## 2. Related Works

Torrens and Benenson argue that the two basic types in models that simulate geographic processes are *cells* and *agents* (Torrens and Benenson 2005). They define relation as a mapping that describes the connection between two entities. As there are two classes of entities, then we have four types of relations: cell→cell, agent→cell, cell→agent and agent→agent. The authors use a leader and follower approach to define the relations. The leader is responsible for managing the relation, while the follower is a passive object. The aim is to keep consistency in the relations, avoiding conflicts between the entities. Therefore we have cell→cell as a spatial neighbourhood, a static relation (two followers). Agent→cell is the cells one agent controls, cell→agent is the agents which belong to a certain spatial location. Finally, agent→agent represents the relations between agents. According to the authors, agent→agent can only be generated by transitivity over the other relations, once the relation has two leaders and it is impossible to define who controls the relation in order to guarantee consistency.

Andrade and others grouped these relations in placements and neighbourhoods (Andrade et al. 2008). Given the two entities, cells and agents, placements connect entities from different types (agent→cell and cell→agent), while neighbourhoods connect entities of the same type (cell→cell and agent→agent). The authors argue that the four relations can be feeded by geospatial data and show that the available toolkits do not support the four types of relation, but they do not present any conceptual framework.

In this work, we consider that, instead of agents having to query a geographic database when they need an answer about the spatial structure, the representation of space within the model is already filled with this data. It represents a suitability map, where the neighbourhoods between cells represents criteria such as visibility or possible movements. We assume the modeller previously knows the queries the agents may perform along the simulation. It can be considered a limitation of this approach, but it is a way to see the problem of using geospatial data, with the advantage of separating GIS

functions from the simulation. The idea is that both applications can work harmonically but separately, sharing only geospatial data.

### **3. GPM as Basis for Setting up Relations**

Based on the fact that Euclidean spaces are not satisfactory to describe the underlying complexity of the relations of the geographic space, Aguiar and others present the idea of Generalized Proximity Matrix, or GPM (Aguiar et al. 2003). The GPM combines data from Euclidian spaces and from topological relations embedded in these spaces to measure spatial relations between geographic objects. The topological relations warp the Euclidean space, reducing distances. One example is the case of transport networks.

There are two strategies for computing distances based on topological relations with GPM. They are called open and closed networks. Open networks can use any location in the edges as entry points, while closed networks use the nodes as the only entry points. The GPM establishes relations between sets of geospatial entities based on geometrical relations. The result is a graph connecting these entities, which can feed each of the four types of relations discussed in the last section. Metrics of distance using Euclidean spaces or networks can be utilized to create neighbourhoods because proximity is the base for these relations when we use the geographic space.

Placements are different from neighbourhoods because the objects must have some overlay. Generally, one object has a fine and the other has a large resolution. In the case of agents→cells, agents have at least a resolution as larger as the resolution of the cells, and the opposite happens in the case of cell→agent. Therefore we can treat placements as multiscale relations.

### **4. The Agent-based Modelling Architecture**

Using the idea of GPM as basis, this section describes an architecture for agent-based modelling of geospatial phenomena. The architecture is part of the TerraME framework. TerraME is a software environment for spatially explicit dynamical modelling (Carneiro 2006). A spatial dynamic model is a model whose locations are independent variables. The outcomes of these models are maps that depict the spatial distribution of a given phenomenon.

The structure of TerraME has a hierarchy of layers, where lower layers provide basic services over which upper layers build new services. The lowest layer is TerraLib, a C++ opensource GIS library (Camara et al. 2008). It provides typical spatial data management and analysis services, and procedures for temporal data handling. TerraLib stores data in layers of information, each one representing a set of geographic objects with the same attributes. It also stores the results of GPM, pointing out the connections between objects and the layer or layers they belong.

The core for the simulation engine and data interface comes directly above TerraLib. From that, TerraME uses a Lua interface to use the concepts designed in the TerraME framework as a Lua extension, with which final users can write their models. Lua is an extensible and high-level scripting language (Ierusalimschy 2003). It has a great acceptance on the games development community due to its simple way to manipulate data and easy interface with other languages. Although Lua is slower than compiled

languages, TerraME implements the hard processing procedures in C++, taking the advantages of both languages.

The main components of the architecture for agent-based modelling are:

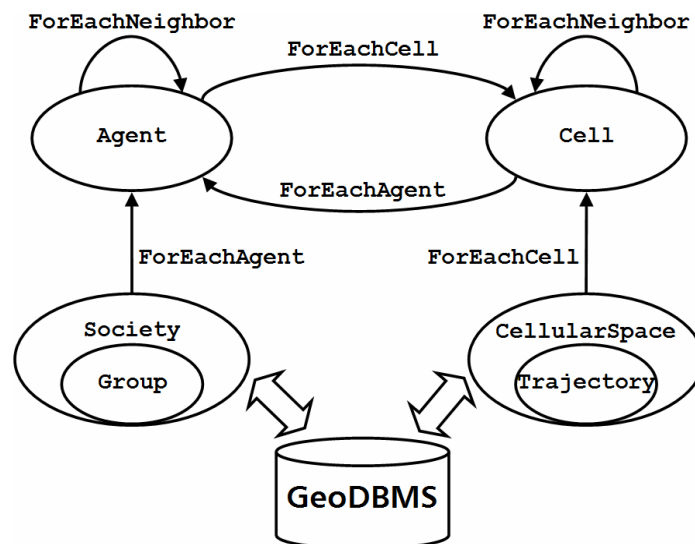
**Agent and Cell:** The basic geospatial entities, each one with its own properties and relations. Agents and cells have unique identifiers to name which objects stored in a layer of a geographic database they represent.

**Society and CellularSpace:** Sets of agents and cells, respectively. Agents within a Society have the same set of properties and general behaviour. CellularSpaces represent cells with the same properties and resolution. Both have a set of properties to create a connection to a database and in which layer of the database the data was read.

**Group and Trajectory:** Ordered subsets of Societies and CellularSpaces, respectively. Both selection and ordering use properties of the entities, therefore they do not consider any geospatial property of the entities.

**GeoDBMS:** A geographic DataBase Management System, in our case a TerraLib database. Data can be loaded as well as written in the geographic database.

Figure 1 shows the main components of the architecture, with the arrows drawing the connections between them. It is possible to use second-order functions to traverse the objects pointed by a given object, applying a function over each of them. For example, `ForEachAgent` can traverse the agents within a Society as well as the agents that belong to a given cell.



**Figure 1. The main components of the architecture and their relations.**

An example of source code for loading a Society, a CellularSpace, and their relations from a given database is shown in Figure 2. The difference between creating a Society and a CellularSpace is the presence of a function to describe the behaviour of the agents. This function, in this case called `createAgent`, receives a data structure containing the properties of a single geospatial entity and returns an agent built using these properties.

Once both objects are created, neighbourhood relations can be loaded directly from them, as shown in Figure 2. In the case of placement relations, as they involve objects from different types, we need a third object of type Environment to encapsulate both in a single object and then connect them.

```
mySociety = Society {
    database = "d:\\database.mdb",
    layer    = "farmers",
    instance = createAgent,
}

myCellularSpace = CellularSpace {
    database = "d:\\database.mdb",
    layer    = "cells",
    select   = {"cover" , "soil_quality"}
}

myEnvironment = Environment {
    mySociety,
    myCellularSpace
}

mySociety:loadNeighbourhood("nearest_farmers")
myCellularSpace:loadNeighborhood("transition_by_roads")
myEnvironment:loadPlacement("farms_to_cells")
```

**Figure 2. Loading entities and their relations from a geographic database.**

## 5. Concluding Remarks

In this work, we presented an architecture where it is possible to use geospatial data stored in a database to create agents, a representation of the space, and the relations involving them. The approach does not focus on urban or ecological applications, but in models with topological relations involving agents and cells. It is possible to create relations directly from geospatial data. With this, the approach of this work is more general than the others available in the literature, once none of them fulfils all the possible ways of using geospatial data.

Currently, we are implementing these ideas in TerraME. This architecture have been applied to develop land use and land cover change models in the Brazilian Amazon. We can cite two models that we have been exploring. The first one is to study the deforestation trajectory in the centre-north region of the Rondônia state (Becker 1997), in Brazil, modelling agents according to their farm size, small and large. There is already a preliminary work which simulates the evolution of deforestation using cellular automata (Carneiro et al. 2004). The objective now is to model using agents that decide what to do with their own sets of cells. The second model will study aspects related to the Açaí market chain in Marajó Island, in Pará state, exploring questions such as its development and sustainability. The açaí occurs naturally in the Amazonian floodplain forest, providing both fruit and the so-called heart of the palm, or palmito (Brondizio 2008). This model will be more complex than the previous one, once it involves agents taking decisions at different scales.

## 7. References

- Aguiar, A. P. D., Câmara, G., and Cartaxo, R. (2003), 'Modeling Spatial Relations by Generalized Proximity Matrices', *V Brazilian Symposium in Geoinformatics - GeoInfo 2003* (Campos do Jordão, SP, Brazil).
- Andrade, P. R., Monteiro, A. M. V., and Camara, G. (2008), 'Entities and Relations for Agent-based Modelling of Complex Spatial Systems', *I Brazilian Workshop on Social Simulation (BWSS'08)* (Salvador, Brazil), 52-63.
- Axelrod, R. (1997), *The Complexity of cooperation* (New Jersey: Princeton University Press).
- Barros, J. (2004), 'Urban Growth in Latin American Cities: Exploring Urban Dynamics through Agent-Based Simulation', (University College London).
- Becker, B. K. (1997), *Amazônia* (São Paulo, Brazil: Ed. Ática.).
- Brondizio, E. S. (2008), *The Amazonian Caboclo and the Açaí Palm*.
- Brown, D. G., et al. (2005), 'Spatial process and data models: Toward integration of agent-based models and GIS', *Journal of Geographical Systems*, 7, 25-47.
- Camara, G., et al. (2008), 'TerraLib: An open-source GIS library for large-scale environmental and socio-economic applications', in B. Hall and M. Leahy (eds.), *Open Source Approaches to Spatial Data Handling* (Berlin: Springer).
- Carneiro, T. G. S. (2006), 'Nested-CA: a foundation for multiscale modeling of land use and land change', (INPE).
- Carneiro, T. G. S., et al. (2004), 'TerraME - A modeling Environment for non-isotropic and non-homogeneous spatial dynamic models development', paper given at LUCC Workshop, Amsterdam.
- Crooks, A. T. (2007), 'The Repast Simulation/Modelling System for Geospatial Simulation', *Working Papers Series - Paper 123* (UCL Centre for Advanced Spatial Analysis).
- Epstein, J. M. and Axtell, R. L. (1996), *Growing Artificial Societies: Social Science from the Bottom Up* (The MIT Press).
- Gilbert, N. (2008), *Agent-based Models*, ed. T. F. Liao (Quantitative Applications in the Social Sciences, 153: SAGE Publications).
- Goodchild, M. F. (2005), 'GIS, Spatial Analysis, and Modelling Overview', in D. Maguire, M. Batty, and M. Goodchild (eds.), *GIS, Spatial Analysis, and Modeling*.
- Ierusalimschy, R. (2003), *Programming in Lua* (Lua.Org).
- Torrens, P. and Benenson, I. (2005), 'Geographical automata systems', *International Journal of Geographical Information Science*, 19 (4), 385-412.