

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

Pedro Ribeiro de Andrade¹, Antonio Miguel Vieira Monteiro¹, Gilberto Camara¹, Tiago Garcia de Senna Carneiro²

¹INPE – National Institute for Space Research, Av. dos Astronautas 1758, CEP 12227-001. São José dos Campos, Brazil
{pedro, miguel, gilberto}@dpi.inpe.br

²UFOP – Federal University of Ouro Preto, R. Diogo de Vasconcelos, 122, CEP 35.400-000. Ouro Preto, Brazil
tiago.carneiro@iceb.ufop.br

Introduction

Simulating social and economic phenomena demands reliable toolkits that allow the user to focus on describing the model, rather than worrying about data structures and simulation control. When we add a spatial component to these systems, with the objective to represent entities and their relations with real-world geographic data, modelling becomes an even harder task, usually requiring strong computational abilities. To make this task easier, we need simple but robust toolkits that generalize the representations of geographic data for creating agent-based models.

This work presents an architecture for building and simulating agent-based models that use real-world geospatial data, taking into account all the ways geospatial data can feed models. Geospatial data can be used to represent space, agents, and their relations (Fig. 1). We focus on how data can be used to create an initial arrangement, as if it was a static representation. We consider that, instead querying a database when the agents need an answer about the spatial structure, the representation of the model already contains this data. We assume the modeller previously knows the queries each agent may perform along the simulation. It separates the simulation from the environment for storing and processing geospatial data, which is already carried out by Geographic Information Systems (GIS). The idea is that both applications can work harmonically but separately.

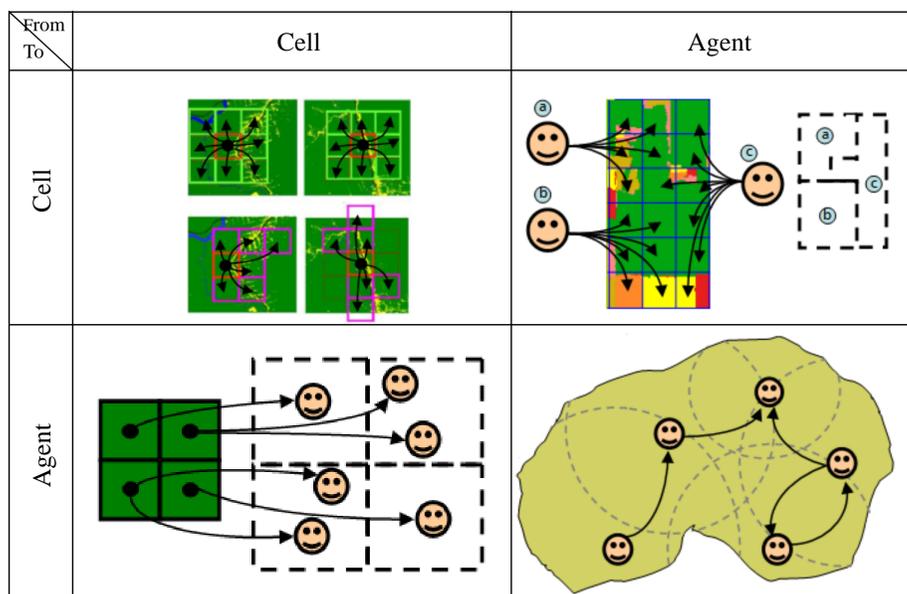


Fig. 1: Examples on how geospatial data can feed relations.

The main hypothesis of this work is that the idea of Generalized Proximity Matrix (GPM) is a foundation for setting up the relations between the entities of agent-based models for simulating geospatial phenomena. GPM is based on the fact that Euclidean spaces are not satisfactory to describe the underlying complexity of the relations on the geographic space [1]. It combines data from Euclidian spaces and from topological relations embedded in these spaces to compute relations between geographic objects. The topological relations warp the Euclidean space, reducing distances.

There are two strategies for computing distances based on topological relations using GPM. They are called open and closed networks (Fig. 2). Open networks can use any location in the edges as entry points. In closed networks, the nodes are the only entry points to the network. GPM establishes relations between sets of geospatial entities based on geometrical relations. The result is a graph connecting these entities, which can be used to feed agent-based models.

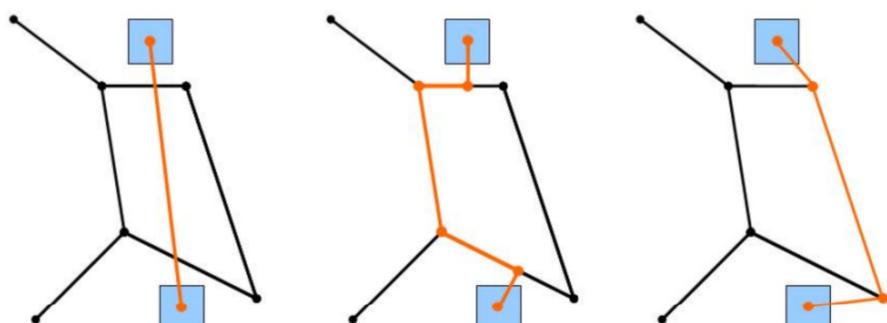


Fig. 2: Strategies for creating relations using GPM. From left to right: Euclidean space, open network, and closed network.

The Architecture

Using the idea of GPM as basis, we have been developing an architecture for agent-based modelling and simulation of geospatial phenomena. The architecture is part of TerraME, a software environment for spatially explicit dynamical modelling based on TerraLib GIS library [2, 3]. TerraLib stores geospatial data as well as the results of GPM in a geospatial database. TerraME provides an interface to Lua, a high-level scripting language [4], with which final users can write their models. Lua has great acceptance on the games development community due to its simple way to manipulate data and its easy interface to other languages.

Fig. 3 shows the main components of the architecture. They are:

Agent and Cell: The basic geospatial entities, each one with its own properties and relations. They have unique identifiers to name which objects stored in 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 spatial resolution.

Group and Trajectory: Ordered subsets of Society and CellularSpace, respectively. Both selection and ordering may use properties of the respective entities.

GeoDBMS: A geographic DataBase Management System, in our case a TerraLib database, to store and retrieve geospatial data.

Captions on the arrows refer to 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.

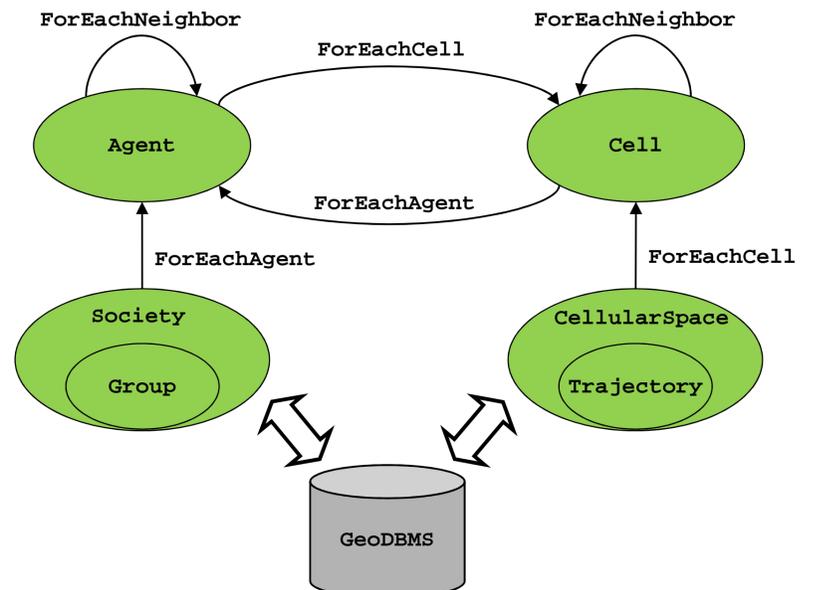


Fig. 3: The main components of the architecture.

Final Remarks

This ABM extension will appear in the forthcoming versions of TerraME. Future works include to use this extension to develop agent-based land use and cover change models for the Brazilian Amazon. More details about TerraME can be found in its webpage [5].

References

1. Aguiar, A. P. D., G. Camara, and R. Cartaxo. **Modeling Spatial Relations by Generalized Proximity Matrices.** in *V Brazilian Symposium in Geoinformatics – GeoInfo'03*. 2003. Campos do Jordão, SP, Brazil.
2. Carneiro, T. G. S., **Nested-CA: a foundation for multiscale modeling of land use and land change.** Doctorate thesis, Image Processing Division. 2006, INPE: São José dos Campos.
3. Camara, G., L. Vinhas, K. R. Ferreira, *et al.*, TerraLib: An open-source GIS library for large-scale environmental and socio-economic applications, in *Open Source Approaches to Spatial Data Handling*, B. Hall and M. Leahy, Editors. 2008, Springer: Berlin.
4. Ierusalimsky, R., **Programming in Lua.** 2003: Lua.Org.
5. TerraME Project. 2009. Available at: <http://www.terralab.ufop.br/dokuwiki/doku.php?id=terralab:terrame>