DYNAMIC NEIGHBORHOODS: A CONCEPTUAL MODEL AND ITS IMPLEMENTATION TO COPE WITH SPATIAL DYNAMICS IN GEOGRAPHIC MODELLING

Raian Vargas Maretto

Proposta de Dissertação de Mestrado do Curso de Pós-Graduação em Sensoriamento Remoto, orientada por Dr. Antônio Miguel Vieira Monteiro e Dr. Tiago Garcia de Senna Carneiro

INPE

São José dos Campos

2010

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Over the 90's, a series of research projects on several geography departments were established based on a systemic view of the urban phenomena. A real city should be properly characterized not just by its fixed infrastructure components. The spatial distribution of different land uses competing and cooperating through interactions and depending on a *neighborhood* definition, established by a spatial proximity criterion, was a more realistic view. Spatial complex dynamics instead of static spatial relations started to be the research focus. This research line has been, since then, very influential over the years. The main instrument devised to investigate these complex spatial dynamics was the use of computer modeling and simulation. It would provide an empirical *test bed* for geographers, urban planners, architects, city engineers, sociologists, and others the like to investigate their theoretical hypothesis and conceptual models of the structure and functioning of urban systems and cities.

Almost at the same time, on the theoretical debate in the geography field on representational perspectives for the geographic spaces, the ideas of *cellular worlds* (COUCLELIS, 1985, 1991, 1997) and a *cellular geography* (TOBLER, 1979) took shape. On the other hand, the use of 2D (bi-dimensional) *Cellular Automata*-CA computational structures in modeling and simulation of complex dynamics in physics and chemistry had already a long tradition. By taking a conceptual view for the geographic space based on *cells* the next step was to start using computer CA-based models to provide the mechanism for observing urban systems dynamics. Much of the works presented on the related literature have taken this road from the 90's until nowadays. It proves that the potential of *Cellular Automata* based models for urban dynamics studies has become widely recognized. (BATTY and XIE, 1994; BATTY, 2000; BATTY, 2005; DEADMAN et al. 1993; PHIPPS and LANGLOIS 1997; WHITE and ENGELEN, 1993, 1997; WHITE et al. 1998; WANER, 1997; ALMEIDA et al., 2003a, 2003b, 2005, 2007; CLARKE et al., 1997, 1998; O'SULLIVAN and TORRENS, 2000; VANCHERI et al, 2008).

In cellular automata (CA) based models, the *neighborhood* definition is a key modeling component, because it determines how the elements that builds up the model interact (Hagoort et al., 2008). It determines how the cells, which represent the geographic objects, are spatially related. To say that two cell-objects are neighbors means that one is exerting some sort of influence over the spatial, temporal and/or behavioral state of the other. The neighborhood relation also brings with it a way of value and differentiate that particular linkage. Usually this is translated into a *weight* mapping definition. The *weight map* represents the strength of the bounds that link those neighbors engaged during the model execution.

Dynamic models aim to represent complex dynamics of real world geographic phenomena. In real world situations, *neighborhoods relations* rarely keep their structure over time. Several studies have demonstrated that the CA-based dynamical models are sensitivity to the neighborhood configuration. Ménard and Marceau, (2005), Kokabas and Dragicevic, (2006) have demonstrated that their land use and urban growth model are sensitive to different cell resolutions and neighborhood configurations. Hagoort et al, (2008) have investigated the impact of using *neighbourhood rules* which are defined and calibrated on an *ad hoc* basis by trial and error methods, and that do not evolve along the model life cycle, on the output of the models.

In an attempt of avoid the problems raised by the use of a fixed *neighborhoods* strategy, a small number of works have been developed more recently. The most recent is the work by Moreno et al., (2009) in which the authors present an implementation of a *dynamic neighborhoods* scheme in a vector-based cellular automata model. In their proposal two objects are neighbors if they are adjacent or separated by other objects in which their states are favorable to the transition of states between them. In this approach, when the objects change its states, the *neighborhood structure* is reconfigured. However, this approach is tightly connected to the very specific CA-extention they are proposing (VecGCA – MORENO et al., 2008). It also does not allow for building *neighborhoods structures* based on the cells attributes or neighborhood rules based on other spatial relations than adjacency.

1.1. This Proposal and its Contribution

This work focus on creating the computational basis for providing *runtime* reconfiguration of the *neighborhoods structures* in CA-based computational modeling frameworks designed to cope with geographic phenomena. Our proposal must allow for the modelers to code *neighborhood reconfiguration strategies* as part of his/her model experiment. In order to do so we propose:

(1) A typology of modeling situations in which a *neighborhood structure* reconfiguration at *runtime* is a requirement from a modeler's perspective. It establishes three categories of changes for the *neighborhood structure* over models life cycle: (a) Endogenous Change, (b) Exougenous Change, and (c) Hybrid Changes.

(2) A conceptual model for defining *neighborhood structures* that change over time based on an algebraic specification that makes use of a graph-like structure. This model provides the basis for defining a set of operators (*functions*) to deal with the problem of allowing truly *dynamic neighborhood structures* for CA-based computational environments.

(3) For evaluation of the concepts proposed in (1) and (2) an implementation will be provided for the **TerraME** (Terra Modeling Environment) platform (CARNEIRO, 2006), an extended CA-based computational framework for spatial dynamic modeling and simulation.

(4) A specific spatial dynamic model will be developed as a *proof-of-concept* experiment. This model needs to incorporate the concept of *dynamic neighborhoods*. The model is to be based on Amaral et al. (2006) that describes the patterns of deforestation and the corresponding evolution of the transportation networks in the Xingu-Iriri frontier, South of Pará State, in the Brazilian Amazon.

1.2. This Proposal and the INPE's Research Agenda on Computational Platforms for Environmental Modelling and Simulation

INPE have been studied the Amazon rainforest, monitoring and measuring it deforestation sate since the 80's. The Amazon region has demonstrated fast dynamics when related to deforestation processes over the years. Recently the interest for studying the human actions in this region has increased considerably. At the INPE the GEOMA project (http://www.geoma.lncc.br/), has set up the INPE's engagement beyond the *Measuring-and-Monitoring* and took the internal research into the complete cycle of *Measuring-Monitoring-Modeling-and-Simulation*. The goal is to build integrated computational models that can observe at multiple scales based on empirical data and process hypothesis. This models should be understood as a computational tool for testing and exploring hypothesis on the dynamics of the social and natural systems over the region.

In a joint effort among INPE and UFOP (Universidade Federal de Ouro Preto – MG), the **TerraME** platform have been developed to model and simulate spatial dynamic phenomena at multiple scales. This platform aims to provide support for institutional projects as GEOMA, to the projects linked to INPE's new CCST – Centro de Ciência do Sistema Terrestre, and any other isolated studies which intend to use a modeling as an empirical possibility for exploring complex dynamical systems. Through the implementation of the *dynamic neighborhoods* capability in this platform, those studies will have a computer plataform that can provide more flexibility for modelers and it expands the universe of problems which can be tackled with **TerraME**.

1.3. Outline of the Proposal

This proposal is organized in 6 sections. In section 2, we present and analyze the related works from the literature. In section 3, we described in detail our proposal. In section 4, we described a model that is used here as a *proof-of-concept* experiment. In section 5, we present the scheduler for his work. And at section 6, some concluding remarks are presented.

2. RELATED WORKS

Over the recent years a series of CA-based computational models for dealing with complex spatial phenomena have been proposed in the literature. Some of them have advanced and have also proposed a computational framework for actually doing the modelling and simulation. All of them acknowledge the importance of the problem of the neighborhood structures definition from a modeller's perspective. Five of these proposed models with a particular emphasis on the specific ways they deal with the modeling of the neighborhood structures are presented and commented here.

2.1. Layered CA model

Straatman et al. (2001) have proposed the Layered-CA model, in an attempt to dealing with multi-scale dynamic models. It is composed by iterative refinement of 2-dimensional cellular automata, in a set of connected layers. Every cell in a certain layer can have one parent in the higher layer and an arbitrary number of child cells in the lower layer, and it composes a vertical neighborhood. It assumes that all layers cover the same area. Figure 2.1 shows this structure, which is similar to a quadtree (FINKEL and BENTLEY, 1974), allowing embedding multiple models operating in different spatial resolutions.

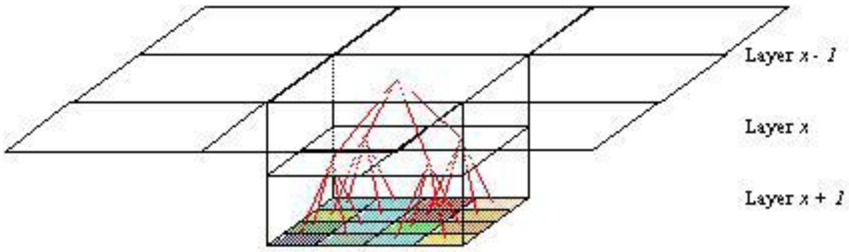

Figure 2.1 – Layered-CA structure, with vertical neighborhood relations. [Source: (STRAATMAN et al., 2001)]

2.1.1. Neighborhood definitions in Layered CA

There are two kinds of neighborhoods in the Layered-CA model, horizontal neighborhoods and vertical neighborhoods. The vertical neighborhoods are composed by the parent and children of a given cell in a given layer. The horizontal neighborhood can be defined by simply boundary conditions, and the neighbors of a cell are the immediately adjacent cells. If we made a selective refinement, as shown in Figure 2.2, all the refined cells can have from 1 to 8 horizontal neighbors. The state of a given cell depends on both, its vertical and horizontal neighbors.

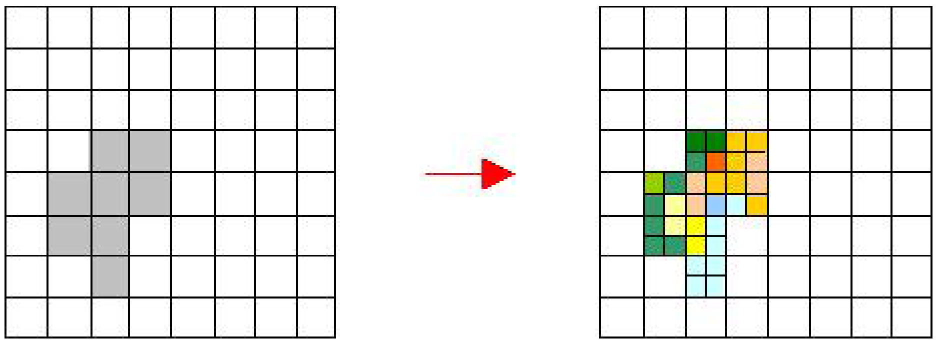

Figure 2.2 – Selective refinement in LCA, where each cell have different numbers of horizontal neighbors. [Source: (STRAATMAN et al., 2001)]

2.2. The DINAMICA model

The DINAMICA model was initially presented for simulation of Amazonian landscape dynamics (SOARES-FILHO et al., 2002; RODRIGUES et al., 2007). Basically, it is a

CA-based model that presents multi-scale vicinity-based transitional functions, incorporating the spatial feedback approach to a stochastic multi-step simulation engine. The calculation of the spatial dynamic transition probabilities can be done by the application of logistic regression. The main input for DINAMICA is a *landscape map*[1], as well as selected spatial variables structured in two cartographic subsets according to their dynamic or static nature. The outputs are simulated landscape maps (one for each time step), spatial transition probability maps, which depict the probability of a cell at a position (*x*, *y*) to change from state *i* to state *j* (being *i* and *j* types of land-use and land-cover class), and dynamic spatial variable maps. Figure 2.3 shows a flowchart of DINAMICA operation.

DINAMICA works in various phases, and to vary the transition and vary the transition rates according to a parameter called *Saturation Value*. The transition rates are passed as fixed parameter within a given phase, and handled by the model when necessary during different dynamic phases. The calculation of the amount of cells to be changed in a given iteration is done by multiplying the number of cells of each land-use and land-cover class, occurring in a time step, by the transition rate. The *saturation value* forces the stopping of the transition *i* – *j* when the number of cells in state *I* reaches a minimum quantity.

The spatial transition probabilities are calculated for each cell and for each specified transition through a polytomous logistic model. Logistic regression is used to depict the probability of a cell to change to one of the states, chosen along the line of the transition matrix. This procedure results in a set of maps depicting the probability of a cell to change to another state.

[1] Land use and land cover map obtained from digital classification of remote sensing image

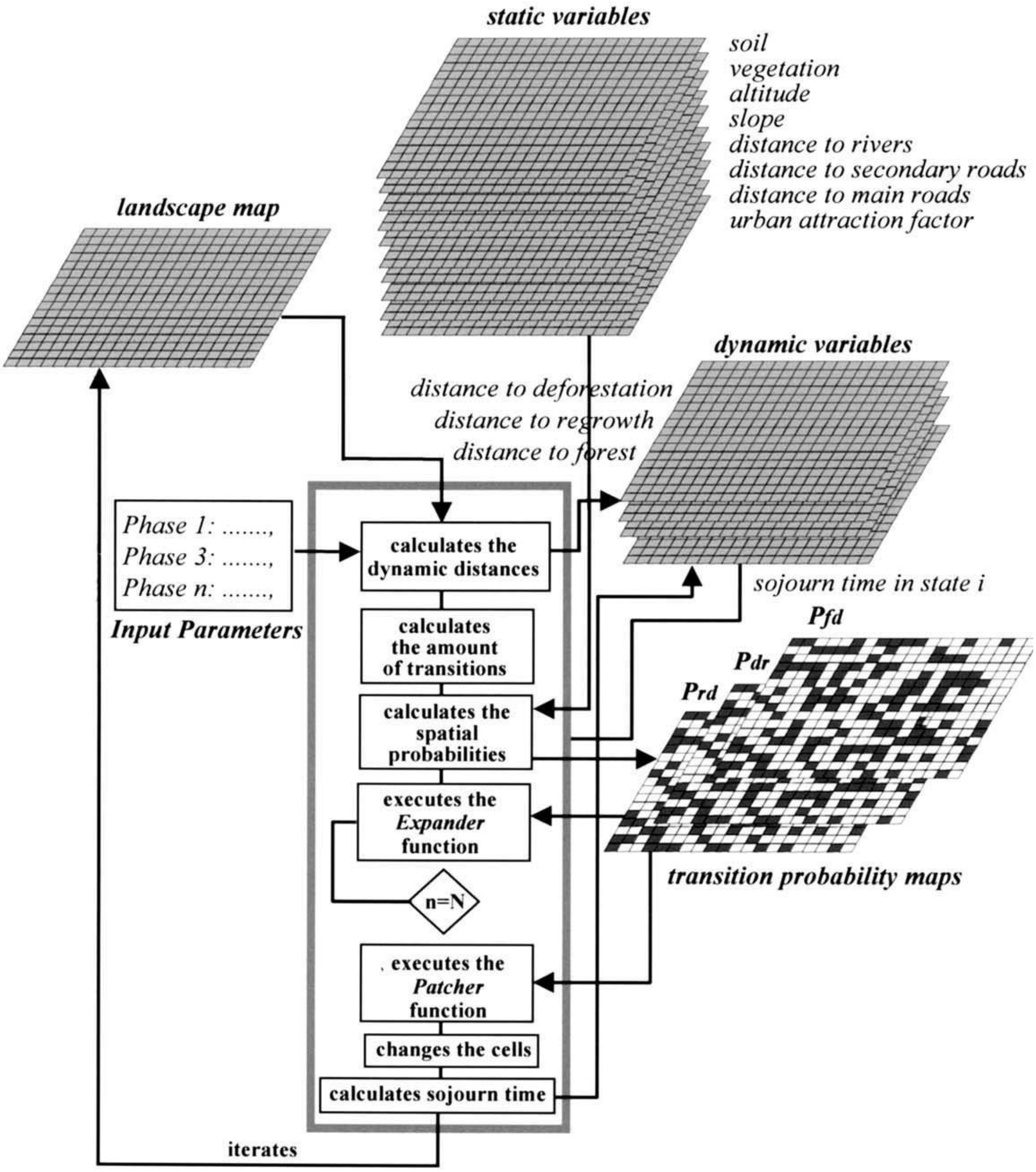

Figure 2.3 – Flowchart of DINAMICA operation. [Source: (SOARES-FILHO et al., 2002)]

2.2.1. Neighborhoods definition in Dinamica EGO – A DINAMICA model implementation

In DINAMICA, the neighborhood influence in the transition probabilities is modelled by splitting the cell selection mechanism into two processes: (a) the *expander function*, dedicated only to the expansion or contraction of previous patches of a certain land-use and land-cover class; and (b) the *patcher function*, designed to generate or form new patches through a seedling mechanism (SOARES-FILHO et al., 2002). The combination of these two processes occurs through the equation below:

$$Q_{ij} = r \times (Expander\ Function) + s \times (Patcher\ Function) \qquad (1)$$

where Q_{ij} represents the total amount of transitions of type ij per simulation step; r and s are, respectively, the percentage of transitions executed by each function, being $r + s = 1$.

To compute the *expander function* the mechanisms provided by the framework considers as cell candidates to be a neighbor only the adjacent cells in a [3x3] connected window.

2.3. Geographic Automata System (GAS)

The *Geographic Automata System* approach (BENENSON and TORRENS, 2004; TORRENS and BENENSON, 2005) combine the concepts of CA based environments and multi-agent systems (MAS), extending it to enable the explicit consideration of space and spatial behavior. It is an attempting in formalizing an object-based view of city structure and functioning within a simulation perspective in urban planning. The GAS concept treats urban infrastructure and social objects as *spatially located automata* named *Geographic Automata* (GA) (BENENSON and TORRENS, 2004). The notions incorporated into the GA are defined in Benenson and Kharbash (2005) as below:

1. An abstract automaton **A** is characterized by a (vector) state **S**, which changes in time according to state transition rules **T**, depending on current input **I**.
2. Cellular Automata theory considers **A**'s neighborhood **N(A)** and assumes that input **I** is defined by an automata belonging to the **N(A)**.
3. Agent **A** of a Multi-Agent System (MAS) can relocate in space, that is, its representation should be capable of managing location and neighborhood changing over time.

The minimal set of transition rules **T** for a *Geographic Automata* **G** should thus specify changes in:

1. Non-spatial attributes of **G**'s state;
2. Location of **G**;
3. Relationships of **G** with other *Geographic Automata*.

Geographic objects and their relationships are then specified as *Geographic Automata* (GA). In GAS, the space is represented as collections of *Geographic Automata*. It also defines an independent set of geo-referencing rules for situating a GA in space. There are two types of GA. *Fixed Geographic Automata* represents objects that not change their location over time. It may be subject to any transition rules, except rules of motion. *Non-fixed Geographic Automata* represents objects that change their location over time, and the full range of transition rules can be applied over it. A GA can be geo-referenced directly or indirectly. Fixed ones are geo-referenced by recording their position coordinates, which does not change over time. Non-fixed GA are geo-referenced indirectly by pointing to other *Geographic Automata*. Figure 2.4 illustrates the two ways of geo-referencing a GA in the GAS model. The structure of the urban space is defined by *spatial relationships* between these set of GA.

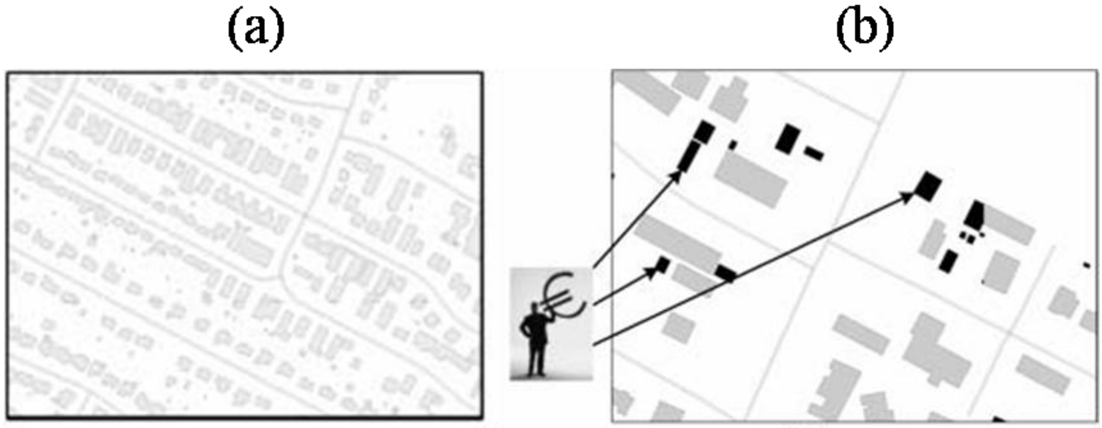

Figure 2.4 – Geo-referencing scheme in GAS (a) Direct – Buildings are represented by means of foundation contours, and roads by means of road boundaries; (b) Indirect – Locating a landowner by pointing to its properties. [Source: Adapted from (TORRENS and BENENSON, 2005)]

2.3.1. Neighborhoods Definition in OBEUS – A GAS model implementation

Object-Based Environment for Urban Simulation - OBEUS, is a system that operationalizes the GAS concept (BENENSON and KHARBASH, 2005). Conceptually, the neighborhoods in a GAS modelling approach can be defined based on the types of the GA. For *Fixed GA*, the neighborhood can be defined by a set of *spatial relationship* possible to capture between *Fixed Objects*. For *Non-fixed GA,* the neighborhood relations are dynamic in space and time. When the geo-referencing rules are based on indirect location, two indirectly located objects can be considered neighbors when the object they point are neighbors, as shown in Figure 2.5. In this case, neighborhood cannot be obtained directly but making use of transitivity as a property of the relations. OBEUS takes these GAS concepts and selects to follow the classical Entity-Relationship Data Model (ERM) (HOWE, 1983) and the Object-Oriented (OO) programming paradigm (BOOCH, 1994) as the main a guidelines for its computational implementation for a GAS-based Modelling platform. Therefore, it considers relationships between entities explicitly. As in the Relational DBMS theory *relationship objects* and their properties are stored in *tables* just as the *objects* representing geographic entities. The neighborhood structure in this case is represented as a set of relationship *objects* linked to specific kinds of GA structures. The idea is to take advantage of the relational-based computational engine by using a *relationship class* and a *table* presentation. With that strategy, OBEUS claims it makes simpler to do on the fly evaluations on whether two GA´s are related, that is to say they are neighbors.

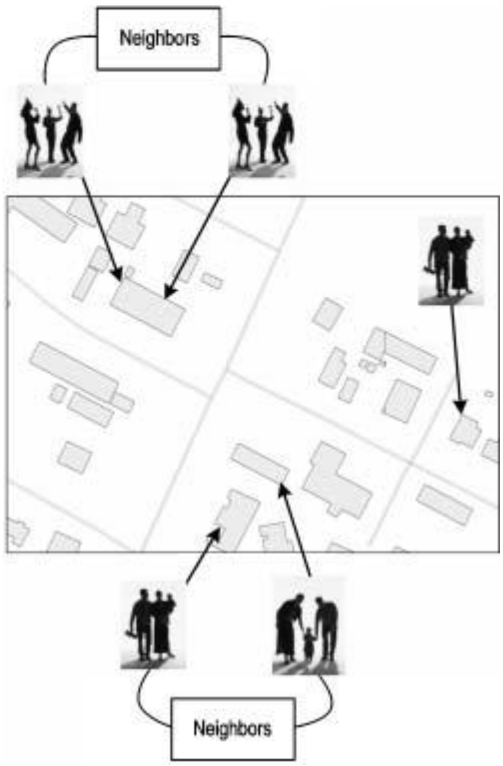

Figure 2.5 – Neighborhood relationships for indirectly GA. Two households are neighbors if they are located in the same property or in neighboring properties. [Source: (TORRENS and BENENSON, 2005)]

However, as the conceptual GAS model is a dynamic system, the properties of a GA as well as their set of *relationships* change over time. In OBEUS, the GAS computational implementation, to avoid inconsistencies, essential limitations are imposed on the semantics of the *relationships*:

1. First, it is assumed that relationships between *fixed GA* are also *fixed*, that is, they do not change once established;
2. Second, it is not permitted *direct relationships* between *non-fixed GA*. That is, *non-fixed GA* can only be related to *fixed GA*. *Non-fixed GA* maintain *relationships* in a transitive fashion. A *non-fixed GA* is related to another *non-fixed GA* only, and if only, they are both related to a specific *fixed GA*. However, *direct relationships* between *non-fixed objects* are undoubtedly important in the real world.

2.4. Nested-CA model

The Nested-CA model was introduced by Carneiro, (2006). Essentially, it is a CA extension oriented to the development of a computational approach for modelling and simulation where an ecological concept of *scale* plays a central role. It sees a *scale* as a set of three fully integrated dimensions: a spatial, temporal and behavioral one. Taking care of them in isolation is not enough to produce the powerful computer representation models one needs for dealing with complex geographical and ecological phenomena. The Nested-CA approach (CARNEIRO, 2006) takes the ecological *scale* concept as the central semantic unit for its computational model. The CA is extended for fully integrated support of the Gibson's *scale* concept (GIBSON et al., 2000). In the Nested-CA, *scales* are taken as building-blocks which can be hierarchically organized. This strategy allows for the development of spatially-explicit models in which *scales* higher in the hierarchy provide overall control to *scales* at the lower levels. Each scale has its spatial dimension represented by one single CA. However, this CA has been extended mainly in three ways: (1) first, it deals with less limited spatial representations, allowing free geometry for the cell-shape definition. This feature makes it possible to work with *cellular worlds* representations coupled with real-world geographical databases that can take *points*, *lines*, *grids*, *polygonal shapes* and even *voxels* structures into modelling (CARNEIRO et al., 2008); (2) second, it allows for a very flexible definition of its cell's neighborhood. Topological cells relationships are expressed in terms of *dynamic neighborhood graphs* (AGUIAR et al., 2000); (3) and third, putting the first two together it can deal with N-dimensional spaces when and where that feature is needed.

Last, but not least important, *Event Schedulers* for dealing with discrete events over a continuous time base are basic components for the *time scale* dimension, and *Infinite-*

State Machines (HENZINGER, 1996) are the building-blocks for the *behavioral scale* dimension. *Event Schedulers* and *State-Machines* are embedded into the Nested-CA macro structure which represents the whole *scale* concept. That is actually what makes the conceptual *scale*-based organization a *Nested* approach: it is nesting a set of extended CA-structures and it is computationally taking care of controlling this built nest fully integrated and making it transparent to the model builder. Figure 2.6 shows schematic representations of a Nested-CA model arrange based on its fundamental *scale* units. A more detailed description can be found in (CARNEIRO, 2006).

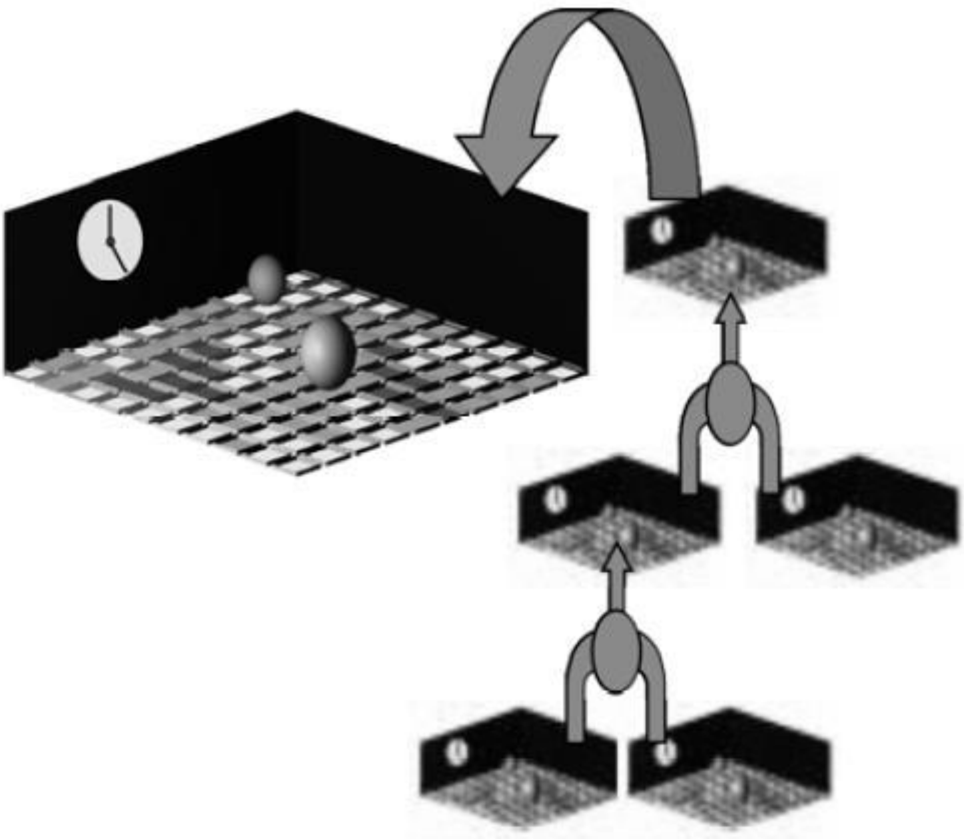

Figure 2.6 – Nested-CA structure [Source: (CARNEIRO, 2006)].

2.4.1. Neighborhoods Structures in the Nested-CA model

The Nested-CA model defines neighborhood for the cells that belong to its *scales* using the *Generalized Proximity Matrices*-GPM concept (AGUIAR et al., 2000). The GPM is an extension of the of the *spatial weights matrix* originally proposed in the context of statistical spatial data analysis (BAILEY and GATTREL, 1995) to be used in the context of spatial dynamic modeling and simulation. The mains idea is to accommodate the notion of *relative space* when defining a neighborhood structure and extend that by using graph-based structures (TAKEYAMA and COUCLELIS, 1997; O´SULLIVAN, 2001a, 2001b). In the Nested-CA, the GPM is the basis for dealing with neighborhood structures definition. Isotropic and non-isotropic neighborhood structures (AGUIAR et al., 2000) can be defined by using the GPM. Figure 2.7 illustrates the use of this concept to define two neighborhoods considering a transportation network and two [3x3] traditional cell-neighborhood structures.

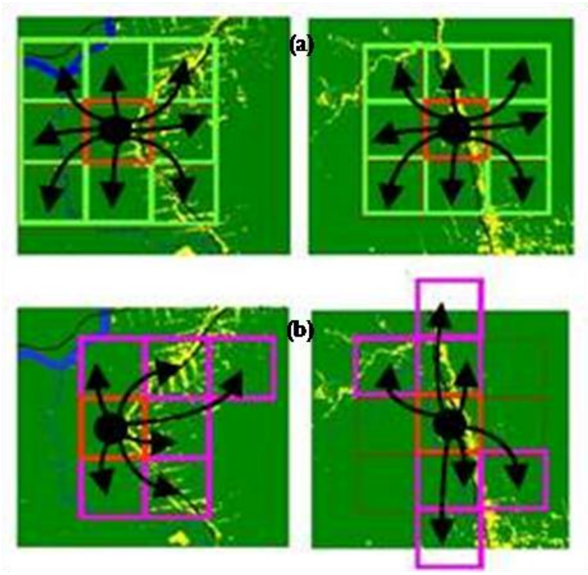

Figure 2.7 – Neighborhoods defined using the GPM concept. (a) Moore neighborhoods. (b) Two non-isotropic neighborhoods defined through a transportation network [Source: adapted from (ANDRADE-NETO et al., 2008)].

However, despite all the flexibility to define the neighborhood structures, once they are defined, they cannot be reconfigured during the model life cycle, in particular they cannot be changed automatically by situations that can pop up at running time triggered by the internal and/or external dynamics of the model in execution. That is to say, following the example of roads shown in Figure 2.7, if a new road is created by an event during the model running time, this change will not be reflected in the future decisions on the next cycle of the model, because the cells neighborhood structure will not change due to this new road included. It needs improvement.

2.5. VecGCA model

VecGCA model is a vector-based geographic CA model presented by Moreno et al., (2008), where the space is represented as a collection of geo-referenced interconnected irregular geographic objects. Each object is in a specific state and can have its proper transition function, according to the area of the neighbor and the influence exercised by it. During the execution of the model, these objects change its shape and size, when a part of its area changes it state.

The transition function is defined to quantify the area of each object that changes state and is evaluated for each neighbor of the object. Such area is related to the area of the neighbor and its influence on the specific object, which is constant over the whole surface of it. This function is equals to 0 when the influence of the neighbor is smaller than a threshold value (λ), which represents the resistance of the object to change its state to the state of the neighbor. The total area of the object is the upper limit, when the whole object changes state.

The geometric transformation reduces the area of the object by removing a quantity from the region nearest to the corresponding neighbor. It is executed n times in each time step, once for each neighbor for which the transition function is greater than 0. It is executed from the neighbor with the highest influence to that with the lowest influence. To perform this transformation, the object is rasterized using a regular grid, which the resolution is defined by the user. The necessary amount of cells changes state to satisfy the area calculated by the transition function.

Thus, the removed cells define a new object which is then combined with the corresponding neighbor. Figure 2.8 shows this process.

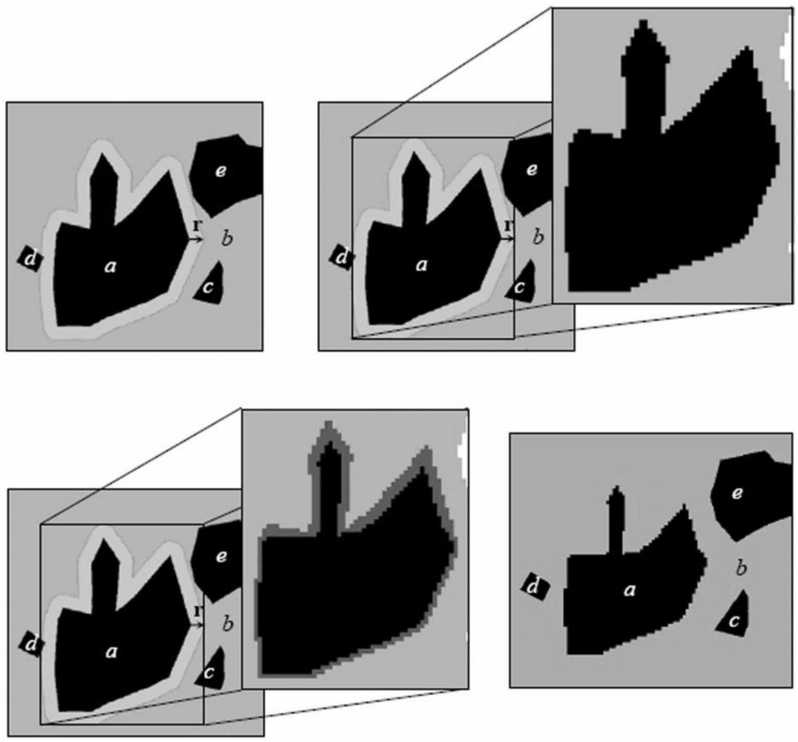

Figure 2.8 – Geometric transformation of an object in VecGCA [Source: (MORENO et al., 2008)].

In the original VecGCA, the neighborhood is defined as an external buffer *d* (region of influence) around each geographic object, and the objects (adjacent or not) that are partially or totally within the region of influence defined by *d* are the neighbors of the central object.

2.5.1. Neighborhoods Definitions in VecGCA

In an attempt to reduce the sensitivity of the VecGCA model to the neighborhood configuration, Moreno et al., (2009) implemented dynamic neighborhoods in the VecGCA model. In this model, two objects are neighbors if they are adjacent or separated by other objects which states are favorable to the transition of states between them. There is no distance or fixed area that delineates it.

The neighborhood includes the whole space, and is specific of each object. A binary matrix as that shown in Figure 2.9 (b) describes if a state *X* is favorable to the transition from the state *Y* to *Z*. The 1 value indicates that the state is favorable to the transition, and 0 indicates the opposite. This matrix indicates, for example, that commercial land is favorable to the transition from undeveloped to developed land, from undeveloped to commercial land, and from park to developed land. Figure 2.9 illustrate this structure.

Thus, we can say, for example, that the neighbors of the object A are the objects B, C, D and E.

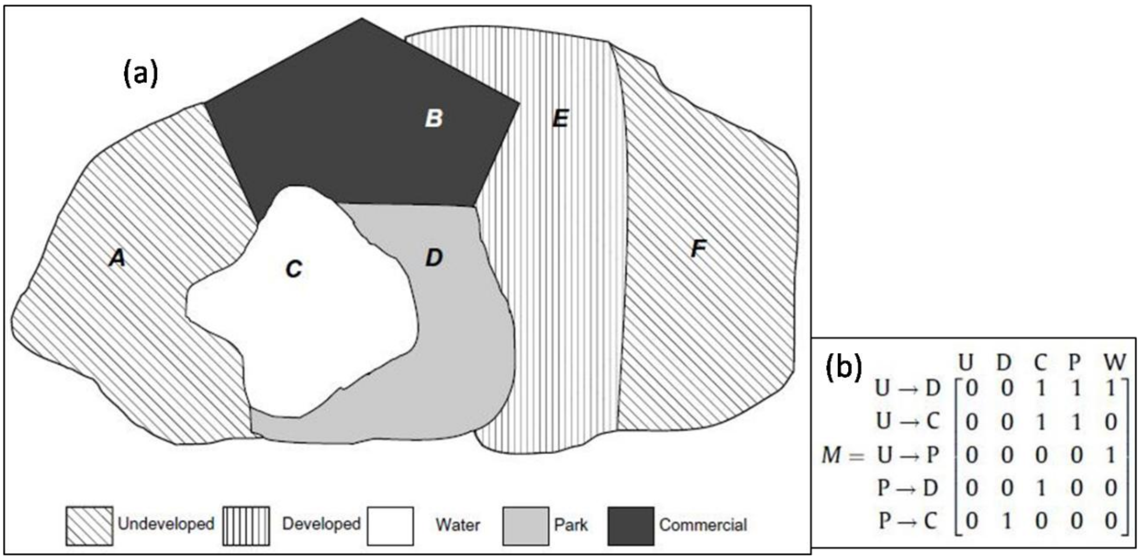

Figure 2.9 – (a) A VecGCA space and (b) the matrix that describes what states are favorable to the transition between what states [Source: Adapted from (MORENO et al., 2009)].

With this neighborhood definition, the influence value is variable on the surface of the central object. Such value increases when the neighbor is closer to the central object, with the maximum value (g_{max}) in the object's border and decreasing inside the object. If g_{max} is higher than the threshold value λ, then the geometric transformation is performed. The transition function calculates a buffer size around the central object which is used in the geometric transformation procedure to take a portion of the central object, adding it to the corresponding neighbor.

The neighborhood structures in VecGCA can be reconfigured at run time but they are entirely dependent on the previously transition table defined for the land-use and land-change model as a whole. The mechanism for changing a specific neighborhood structure is part of the running model as an adjust of neighborhood relations once objects in the model have made their transition from one state of land use to other state of land use. It does not produce a general mechanism for dealing with changes in neighborhood structures over space and time, it has its use strict to models based on transition-sate tables.

3. THE PROPOSAL

Over this section a conceptual model for defining neighborhood structures that change over time is presented. This model is an algebraic specification based on a Graph-like structure aimed to provide the basis for an implementation of a set of operators (functions) to deal with the problem of allow truly dynamic neighborhood structures definition for CA-based computational environments designed to geographic modeling and simulation. As a *proof-of-concept* and for evaluation of the proposed conceptual model, an implementation of it will be made in **TerraME** – *Terra Modelling Environment* (CARNEIRO, 2006), a computer platform that implements the Nested-CA model (http://www.terralab.ufop.br/dokuwiki/doku.php?id=terralab:terrame). This implementation will be used to develop a local LUCC-*Land Use and Land Cover Change* model that incorporates the concept of spatial dynamic neighborhoods. The model is to be based on Amaral et al. (2006) that describes the patterns of deforestation and the corresponding evolution of the transportation network in the Xingu-Iriri frontier, South of Pará State, in the Brazilian Amazon. The ideas for this model are presented in section 4.

3.1. Dynamic Neighborhoods (DN): A Conceptual Model

3.1.1. Defining the GPM-*Generalized Proximity Matrices*

Based on the idea that spatial relations in geographic space are more complex than simply the Euclidean space metrics currently most used and available in GIS systems, Aguiar et al., (2003) proposed the concept of a *Generalized Proximity Matrices (GPM)*. The GPM is an extension of the of the *spatial weights matrix* originally proposed in the context of statistical spatial data analysis (BAILEY and GATTREL, 1995) to be used in the context of spatial dynamic modeling and simulation. The main idea is to accommodate the notion of a *relative space* when defining a neighborhood structure, and extending that by using graph-based structures (TAKEYAMA and COUCLELIS, 1997; O´SULLIVAN, 2001a, 2001b). It combines the notion of space as a set of *absolute locations* in a Cartesian coordinate system, named *absolute space*, with the notion of space as a set of *spatial relations*, which are dependent on *topological connections* and *fluxes* between physical or virtual networks, named *relative space*.

The GPM allows the extension of spatial analysis formalisms and techniques to incorporate relations on *relative space*, providing a new way for exploring complex spatial patterns and non-local relationships in CA-based dynamic models. Recent developments have extended the GPM approach positioning it as a generic way of expressing spatial relations between geographic objects such as cells and agents (MOREIRA et al., 2008, 2009).

Formally, a GPM can be defined as follows:

Being a set *O* of spatial objects whose geometrical representations are defined over a connected subset $S \subset \Re^2$. Given two objects o_i and o_j belonging to *O*, the proximity relation between than is denoted w_{ij}. The GPM is defined as a set *W* of triplets $[(o_i, o_j, w_{ij})]$, which each pair of element is associated to a combined proximity measure. Additional information about the network relations between the objects in *O* are given by a graph *G* over subset *S*.

The computation of each element w_{ij} of the GPM requires two proximity measures, defined by the functions *proxabs* and *proxrel*, which are associated to absolute space and relative space, respectively. Basically, each w_{ij} is calculated as following:

$$w_{ij} = proxabs(o_i, o_j) \cup proxrel(o_i, o_j) \qquad (1)$$

3.1.2. An Algebraic Definition for Dynamic Neighborhoods Structures

Consider the set *O* of spatial objects described above and a set *C* of cells in which each cell $c_i \in C$ represents an object $o_i \in O$. The set *G* is a set of graph-like structures which represent the relations between the elements from *C* and, consequently, of the elements from *O*. *W* is the set of weights of the relations between the cells (or objects) and each element $w_{ij} \in W$ is the weight of the relation between c_i and c_j. In order to define the following concepts, the equation (1), above will be taken as the function $f_w \in FW$, where *FW* is the family of functions for computing w_{ij} (union of the functions *proxabs* and *proxrel*). From the *FW* family another family of functions *F* is defined where a specific function $f_{NL} \in F$ for each pair of cells (c_i, c_j) establishes a *neighborhood law* for the entire *CS set*, defined as the *Cellular Space* that contains all the cells c_i from *C*.

Having defined these *sets* and *functions* above, we will introduce a typology of modeling situations in which a neighborhood structure run-time reconfiguration is a requirement from a modeler's perspective to produce more adherence between the modeler's knowledge of the phenomena dynamics and his/her possibilities of coding that into the executable model.

1. ***Endogenous change of the neighborhood structure:*** Consider we have a spatial process $p_i \in P$, which represents a geographic phenomenon. This process is represented by the model $m_i \in M$. During m_i life, the modelling structures changes, creating new objects or changing existing ones. These changes are made by a m_i structure modeling. Once it cause changes in the neighborhood of a cell c_i, the graph g_i needs reconfigurations, to ensure that the neighborhood follow the model evolution. It is called *endogenous* because the forces causing changes came from inside m_i, i. e., from inside the spatial process p_i being modelled. Figure 3.1 shows a scheme of this.

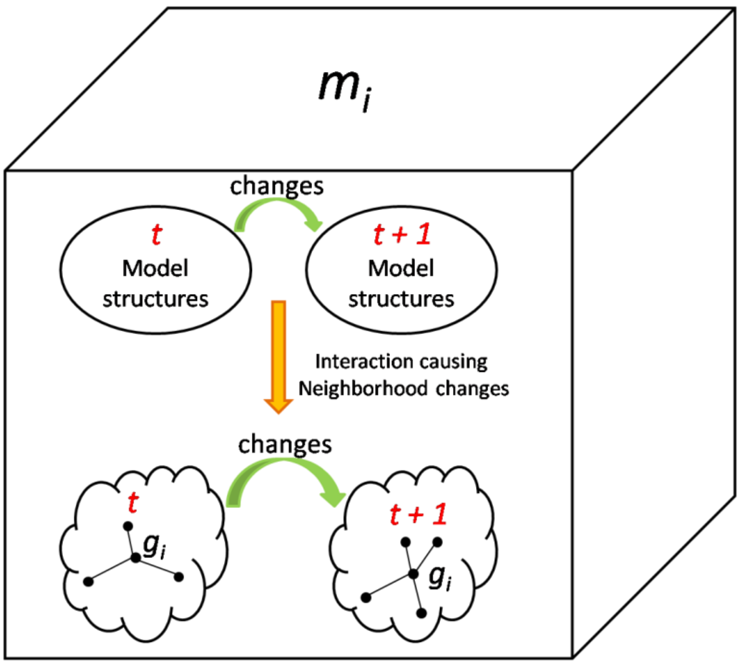

Figure 3.1 – Scheme of the endogenous change of the neighborhood structure.

2. ***Exogenous change of the neighborhood structure:*** Consider we have the set $P = \{p_1, p_2,...,p_n\}$ of spatial process being modelled by the set of models $M = \{m_1,m_2,...,m_n\}$, where each $m_i \in M$ models a corresponding process $p_i \in P$. The models from M are running in parallel and interacting between them. At a moment, a change that occurs in a model m_j may cause a change in the dynamic behavior of a model m_i. This change may be in the neighborhood of a cell c_i from the model m_i, and the graph g_i, which represents it

neighborhood needs reconfigurations. It is called *exogenous* because the forces causing changes came from outside the m_i, i. e., from outside the spatial process p_i being modelled by m_i. Figure 3.2 shows a scheme of this.

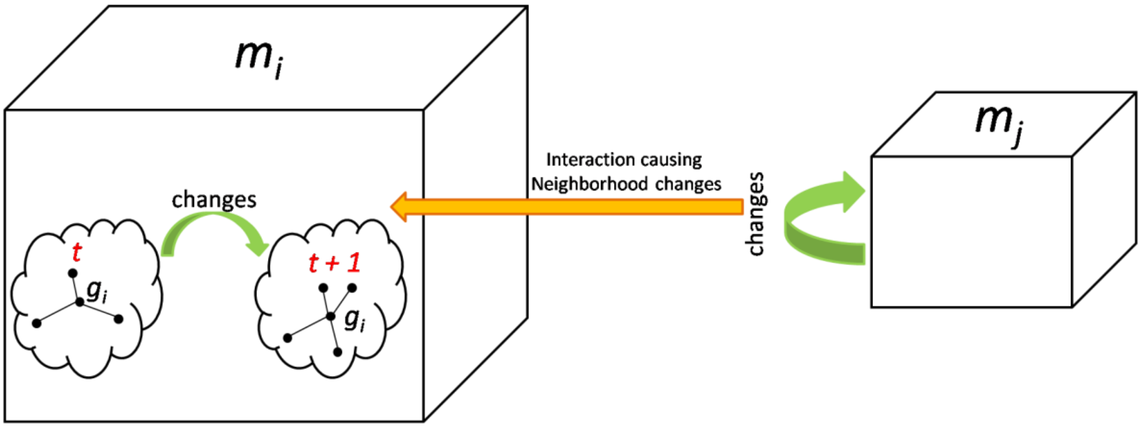

Figure 3.2 – Scheme of the exogenous change of the neighborhood structure.

3. ***Hybrid change of the neighborhood structure:*** This type is the most common and therefore, the most important case. In this case, we have the two previous cases occurring in parallel. Considering a set P of spatial process and a set M of models modelling the processes from P. At a moment, changes occur in the modelling structures from a model m_i and in the structures from another model m_j. The function $f_{NL} \in F$ used to define the neighborhood of an object c_i, represented by the graph g_i, from m_i is defined considering both, the modelling structures from m_i and from m_j. When the changes occur in the structures from these models, both reflects in the neighborhood of c_i, and g_i need to reflect these changes. It is called *hybrid*, because the forces causing changes came from inside and from outside m_i, i. e., from inside and from outside the spatial process p_i being modelled by m_i. Figure 3.3 shows a scheme of this.

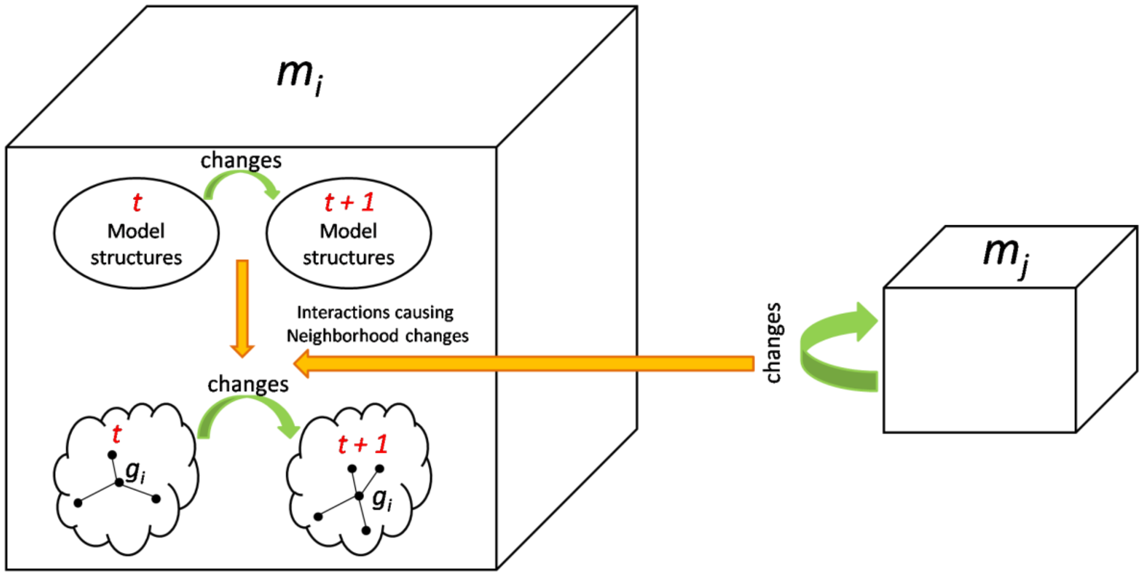

Figure 3.3 – Scheme of the hybrid change of the neighborhood structure.

For providing truly dynamic neighborhood structures reconfiguration at model run-time, all we need to do is to define a set of basic functions over those defined sets. Thus, keep in mind that they shall provide the means for dealing with the three (3) modeling situations described by the typology proposed here.

In order to do so, we need first to establish a set of atomic operators. These functions must operate at the cell, $c_i \in C$, level:

1. ***Add: CxWxG→G*** denotes the function that adds a cell c_j in the neighborhood of another cell c_i. Basically, c_j becomes a new node in the graph $g_i \in G$ with a weight w_{ij}, which represents the strength of the relation. After add c_j in g_i, w_{ij} is added into the GPM structure, ***G***.
2. ***Remove: CxG→G*** is a function that removes the cell c_j from the neighborhood of another cell c_i. Basically, the node corresponding to c_j in the graph $g_i \in G$ is removed, and w_{ij} is removed from the GPM structure, ***G***.
3. ***SetWeight: CxWxG→G*** is the function that sets the weight w_{ij} of the relation between the cells c_i and c_j. Basically, it changes the weight of the node that represents the cell c_j in the graph $g_i \in G$ and changes the value of w_{ij} into the GPM structure, ***G***.
4. ***GetWeight: CxG→W*** is the function that given two cells c_i and c_j, returns the weight w_{ij} of the relation between them. Basically, it returns the weight of the node corresponding to the cell c_j in the graph $g_i \in G$.

From these small set of atomic operators we need to build more complex operators, which will allow for composed operations that act on the *CS-Cellular Spaces* type of sets:

5. ***NeighborhoodLaw: CSxCS→Boolean*** is the function that defines the condition for considering two cells neighbors. Usually it is defined when creating the neighborhood, and don't change during the model life cycle. It receives two cells c_i and c_j and return "true" if they satisfy the condition and "false" otherwise. This condition could be defined by two ways:
    a. over the set $A = \{(A_1,\leq), (A_2,\leq),\ldots, (A_n,\leq)\}$ of partially ordered domains of cells attributes, where a_i is a value of the attribute $(Ai,\leq)$, i. e., $a_i \in (A_i,\leq)$;
    b. over spatial operations. For example: given two cells c_i and c_j, they are neighbors if c_i "touch" c_j.
6. ***Reconfigure: FxFWxCSxG→G*** is the function that reconfigure the whole neighborhood graph g_i of the cell c_i from instances of functions of the families *F* and *FW*, where *F* are *NeighborhoodLaw* functions. The traversal of the *Cellular Spaces* $cs_1,cs_2,\ldots,cs_n$ from *CS* is done by applying the function $f_{NL} \in F$ for each pair of cells (c_i,c_j), where c_i is the central cell of the g_i. For each c_j, it executes:

$$\begin{cases} Add(c_j, w_{ij}, g_i), & if\ f_{NL} = "true"\ and\ c_j \notin g_i \\ SetWeight(c_j, f_w(c_i,c_j), g_i), & if\ f_{NL} = "true"\ and\ c_j \in g_i\ and\ f_w(c_i,c_j) \neq w_{ij} \\ Remove(c_j, g_i), & if\ f_{NL} = "false"\ and\ c_j \in g_i \end{cases} \quad (2)$$

To apply the ***Remove*** function to the neighborhoods of all the cells of a *Cellular Space-CS*, we need to traverse it applying the function for each one of the cells.

Thus, the changes of the neighborhood structure can occur in two forms, which even can occur simultaneously:

- Changing the geometry of g_i, i. e., changing the number of arcs and nodes;
- Keeping the geometry of g_i, i. e., changing only some values of w_{ij}.

3.2. Dynamic Neighborhoods (DN): The Implementation Model

For evaluation of the proposed conceptual model an implementation of it will be made in **TerraME** – *Terra Modelling Environment*, a computer platform that implements the Nested-CA model. To describe the implementation model, first the TerraME platform is described and, after that, it is pointed a possible signature for the functions described before and an overview of how they will be inserted in **TerraME**.

3.2.1. The TerraME platform

TerraME-*Terra Modelling Environment*- is an open source software platform which implements the Nested-CA model and services for spatiotemporal data analysis and management, for building spatially explicit dynamical models (CARNEIRO, 2006). The **TerraME** Programming Language is an extension of the Lua programming language (IERUSALIMSCHY et al., 1996), a powerful and expressive language specially designed for extending applications. The source code of a Model in **TerraME** can be written in a text editor, and it is interpreted by the **TerraME** interpreter. Figure 3.4 shows a scheme of the **TerraME** Programming Environment.

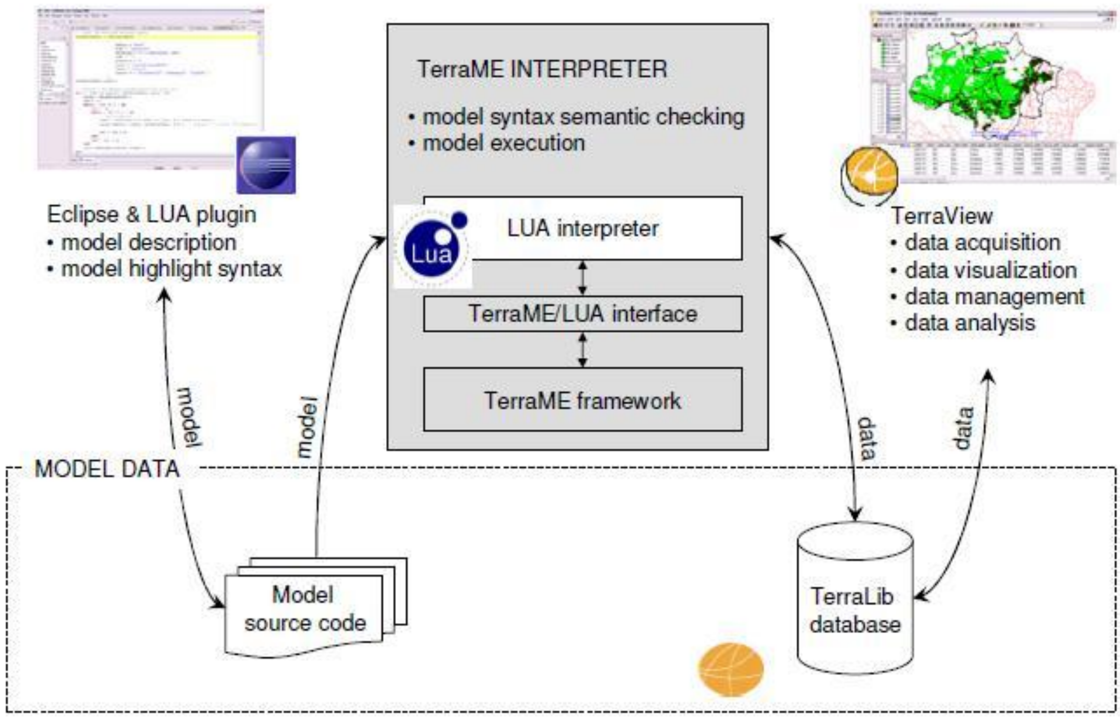

Figure 3.4 – The TerraME Programming Environment [Source: (CARNEIRO, 2006)].

The inputs of the models are spatial data stored as GIS layers or in spatial data file formats. The outputs are *maps* that describe the spatial distribution for the variables involved into the phenomena being modelled. To support spatial data and spatial operations, **TerraME** is coupled with the **TerraLib** GIS library (CÂMARA et al., 2008), which provides services for model data input and output, data storage and spatial data management and operations. The input data and the outputs of the model can be visualized, manipulated, and analyzed in **TerraView**, a **TerraLib**-based GIS.

Like many other toolkits, **TerraME** was implemented in a *layered architecture*, shown in Figure 3.5. In this structure, the components of each layer implements its services calling the services of lower layers. It makes easier to reuse the layers and keeps locally the dependences.

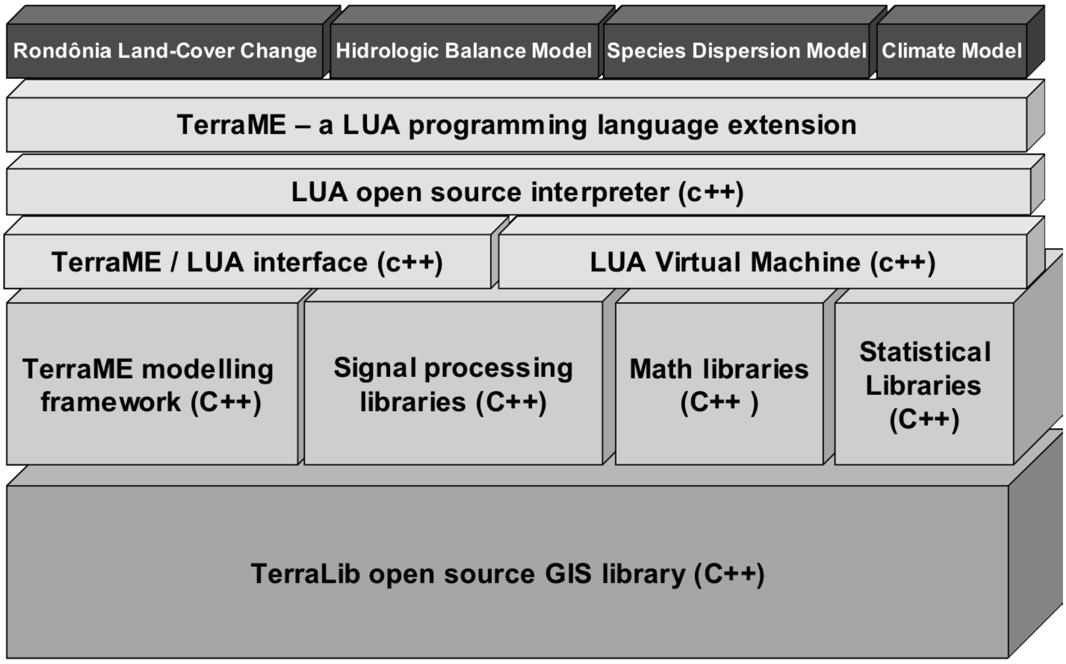

Figure 3.5 – TerraME Software Architecture [Source: (CARNEIRO, 2006)].

Conceptually, the TerraME modelling framework is divided in three modules: *spatial*, *temporal* and *behavioral*. The temporal module implements the scheduler, i. e., all the functions that control the time of the model. The behavioral module implements all the functions that control the behavior of the components of the model, i. e. the state machines that control the behavior of each cell (local) and of the whole (global). And finally, the spatial module implements all the function that creates and manipulates the geographic space in the model, i. e., the cellular spaces and its functionalities, the cells and its functionalities, the trajectories and the neighborhoods. In this work, as we are proposing improves to the neighborhoods structures, the implementation of this work will be done in the spatial module.

3.2.2. The dynamic neighborhoods implementation

To incorporate the concept of dynamic neighborhoods in the **TerraME** modelling framework, we will implement the functions and structures described over the section 3.1.2 by using Lua. The functions signature and their operating mode are described in this section. Most of these functions will be implemented first into the **TerraME** kernel as functions written in C++ and after, they will be available for the Modeler's as Lua functions inserted into the TerraME programming language.

Each *neighborhood object* is related to a *cell object*, and each *cell object* can have any *neighborhood object* related to it. The *neighborhood object* stores and control who are its neighbors, which are the *weights* for each relation, and what is the *neighborhood law* used to create it. The atomic functions previously described: *Add, Remove, SetWeight,* and *GetWeight*, which will be used to manipulate the *neighborhood objects*, have the following signatures and operations:

1. ***Neighborhood.addNeighbor(cell, weight)*** → this function receives a *cell object* and a *weight* (a real number) as parameters. It adds this cell and this weight to a *neighborhood object* of another cell.
2. ***Neighborhood.removeNeighbor(cell)*** → this function receives as parameter a reference to a *cell object*, and removes it from a *neighborhood object* from another cell.
3. ***Neighborhood.setWeight(cell, weight)*** → this function receives as parameter a reference to a *cell object* and a *weight*, and replaces the weight stored in the *neighborhood object* by this new one.
4. ***Neighborhood.getWeight(cell)*** → this function receives as parameter a reference to a *cell object* and returns the weight stored in the *neighborhood object* for it.
5. ***Neighborhood.isNeighbor(cell)*** → this function is an auxiliary function that receive as parameter a reference to a *cell object*, and verifies if it is in the current neighborhood, returning "true" if it is and "false" otherwise.

The *neighborhood law* and the f_W (here called *fWeight*) functions, which will be used in the *reconfigure* function, will not be implemented, because it will be created by the modeler, when constructing the model. But, these functions have a predefined signature, which need to be defined as follow:

6. ***Boolean neighborhoodLaw(cell_1, cell_2)*** → This function receives two cells as parameter. It needs to define a condition, by which these two cells will be considered neighbors or not. If these cells satisfy this condition, the function returns “true”, otherwise, return “false”.
7. ***Double fWeight(cell_1, cell_2)*** → This function receives two cells as parameter. It defines how the *weight* of the relation between then will be computed. It returns a real number, the *weight* of the relation.

The *reconfigure function* will have the following signature:

8. ***Neighborhood.reconfigure( )*** → this function receives no parameters. It goes through all *Cellular Spaces* related to the current neighborhood being reconfigured, applying the *neighborhoodLaw( )* function for each cell. For this way, the *reconfigure( )* function verifies what cells satisfy the condition defined by the *neighborhood law*. If the current cell satisfy the condition, first apply the *isNeighbor*( ) function, to verify if it is a neighbor in the current neighborhood or not. If the *isNeighbor( )* function returns “false”, just add the current cell in the structure with the *weight* computed by the *fWeight( )* function, using the *add( )* function. If it returns “true”, just adjust the weight of the relation, changing the current weight by that calculated by *fWeight( )*. If the *isNeighbor( )* function returns “false”, do nothing. If the current cell do not satisfy the condition, either apply the *isNeighbor( )* function. If it function returns “true”, just remove the current cell from the current neighborhood, using the function *remove( )*. If return “false”, do nothing. Figure 3.6 presents the pseudocode of the *reconfigure( )* function, where *CS* is a cellular space, *cell* is the “central” cell of the neighborhood, i. e., the cell for which the neighborhood will be reconfigured, and $cell_i$ is the *i-th* cell during the traversal of the *Cellular Spaces*.

```
cell.Neighborhood.reconfigure( )
    For Each Cell from (cs_1, cs_2,…, cs_n)
        IF <neighborhoodLaw(cell, cell_i ) = "true"> THEN
            IF <isNeighbor(cell_i) =  "false"> THEN
                add( cell_i, fWeight(cell, cell_i) )
            ELSE
                setWeight( cell_i, fWeight( cell, cell_i) )
            END
        ELSE IF <neighborhoodLaw(cell, cell_i) = "false"> THEN
            IF <isNeighbor(cell_i) = "true"> THEN
                remove(cell_i)
            END
        END
END
```

Figure 3.6 – Pseudocode of the *reconfigure( )* function, where: *CS* are *Cellular Spaces* and $cell_i$ is the *i-th* cell during traversal of the *Cellular Spaces*.

In addition to these functions other auxiliary functions will be needed, which will be used in the changes caused by the interaction between the models, in the *exogenous* and *hybrid* modes of operation, and to make possible future analysis of the neighborhood structures and its evolution during the model execution. These functions will promote the persistence of the neighborhoods, which can be done by two ways: in a file system, or in a database scheme. The auxiliary functions that will be implemented have the following signatures:

1. ***Cell.loadNeighborhood( )*** → loads a full neighborhood structure from a chosen database.
2. ***Cell.saveNeighborhood( )*** → saves a full neighborhood structure in a chosen database.
3. ***Cell.loadNeighborhoodFile( )*** → loads a full neighborhood structure from a chosen file.
4. ***Cell.saveNeighborhoodFile( )*** → saves a full neighborhood structure in a chosen file.
5. ***Cell.loadChanges( )*** → loads just a part of a neighborhood structure from a chosen database. It is important, because in some cases, we have just few changes between two iterations, and it is better to store just the changing parts than to store the full structure. Saving the full structure in these cases, we could have redundancy in the stored data.
6. ***Cell.saveChanges( )*** → saves just a part of a neighborhood structure in a chosen database.

In addition, another function can be defined and which will be such useful in the analysis of the changes that occurs in the neighborhood structures. It enables to show graphically the neighborhood graph of a given *cell object*, making possible better analyses of its dynamics.

7. ***Cell.graphDisplay( )*** → shows in an graphic interface the neighborhood graph which represents the neighborhood of a given *cell object*. It provides graphic representations for the weight and direction of each neighborhood relation.

After implementing all these functions, we can define their use in terms of implementing code for models that fit the typology proposed previously.

1. ***Implementation of the endogenous change of the neighborhood structure:*** In this case, just the functions using internal rules to manage the reconfiguration of the neighborhoods are used, i. e., the functions *addNeighbor( ), removeNeighbor( ), setWeight( )* and *reconfigure( ).* Using these functions, the modeler can define rules, through which the model will internally reconfigure the neighborhoods. Figure 3.7 shows a scheme of how it will operate.

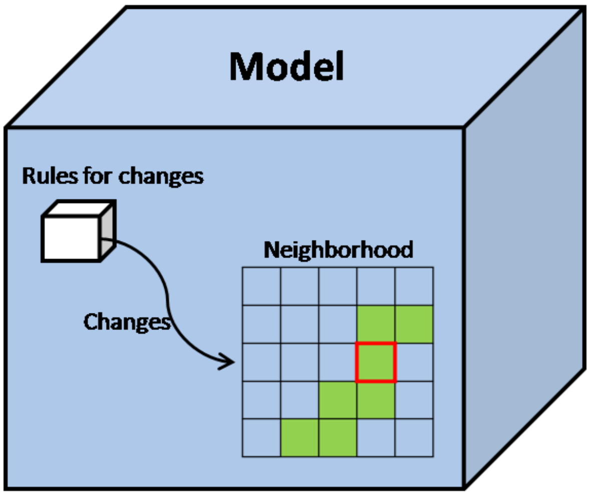

Figure 3.7- Operational scheme of the endogenous change of the neighborhood structures.

2. ***Implementation of the exogenous change of the neighborhood structure*:** In this case, not only the functions manipulating internal rules are used, but either the functions used to manage external data, i. e., data

coming from another models. TerraME enables the execution of several models in parallel and interacting between them, through the *Environment* concept (CARNEIRO, 2006). Considering there are two models executing in parallel, and changes occurring inside the model 2 cause changes in the neighborhood structure of certain objects of the model 1. When it happen, the model two save the neighborhood structure, or just the changes caused by it action, and shoot a trigger warning model 1, who load the new structures. Another case occurs when the models are not running in parallel, but the operation is similar. When model 2 execute, it save the structures for each time, and model 1, when executing, load this structures for each time it is necessary. Figure 3.8 shows a scheme of how it operates.

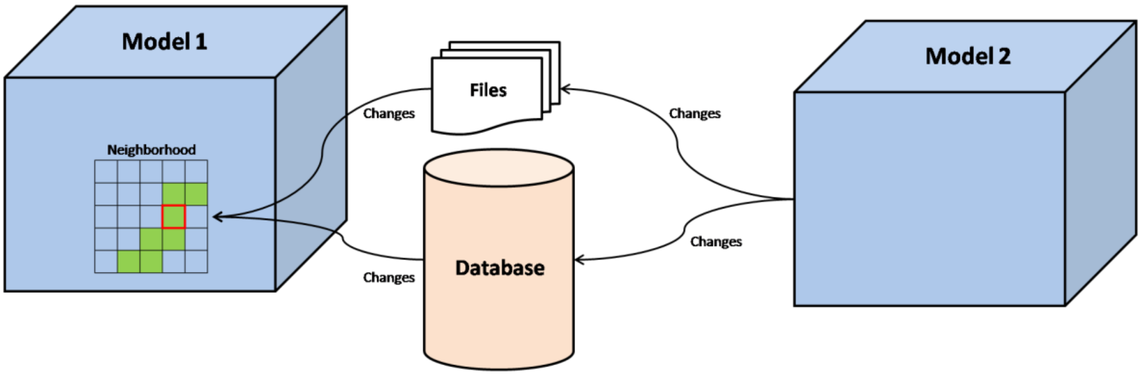

Figure 3.8 – Operational scheme of the exogenous change of the neighborhood structures.

3. ***Implementation of the hybrid change of the neighborhood structures***: In this case, we either will use both, the functions manipulating internal rules and the functions to manage external data. However, here the two cases described above (endogenous and exogenous) will occur simultaneously over the same structure. Figure 3.9 illustrate the operational scheme of this case.

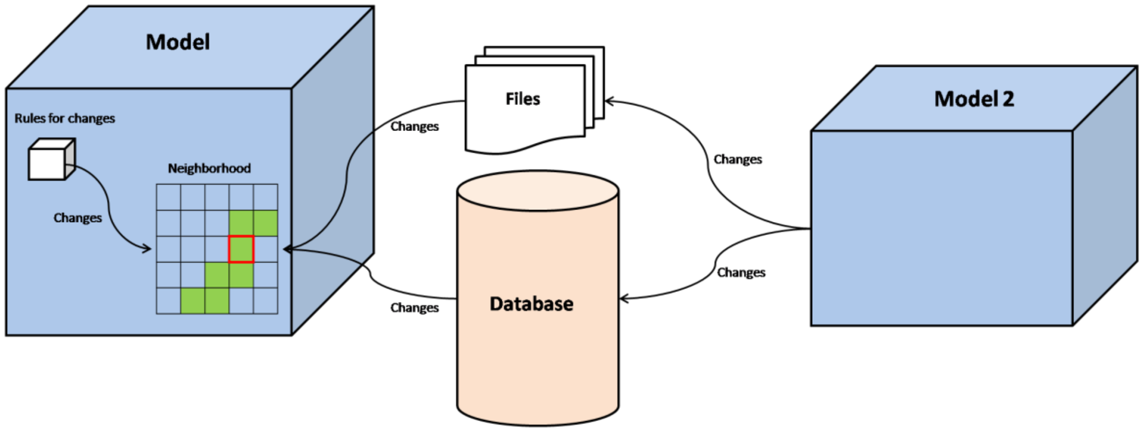

Figure 3.9 – Operational scheme of the hybrid change of the neighborhood structures.

4. A *Proof-of-Concept* Experiment: Modelling Deforestation Patterns Through Circulation Networks in an Amazonia Frontier Site

In order to prove the concepts proposed in this work, we will develop a LUCC (Land Use and Land Cover Change) model that needs the dynamic neighborhood concept. Based on Amaral et al. (2006) we will simulate the deforestation patterns in the Xingu-Iriri frontier, South of Pará State, in the Brazilian Amazon, over a period that goes from 2000 to 2009. There are two main reasons for choosing this case study to be our *proof-of-concept* experiment: (1) the fact that there is previous studies at INPE that are still going on (ALVES et al, 2010; AMARAL et al., 2009; ESCADA et al, 2005; AMARAL et al, 2006) describing the process of land cover change and its relation to the evolution of the transportation networks in that region; (2) the availability of an already built geographical database.

Basically the model aims to analyze land cover change patterns in the region, taking into account the *circulation networks*[2] as a major influence. These networks play a fundamental role in new frontiers, amplifying existing tendencies and intensifying certain process linked to observed deforestation patterns. *Circulation networks* can accelerate the migratory process, as well as the local economic activities and they play a role on local deforestation process. Instead of concentrating the modeling process on the *roads-only-induced-deforestation* approach this region allows for looking at the coupled

[2] Circulation networks are multi-modal transportation networks composed by roads, rivers and runways

dynamics linking the observed deforestation patterns and the evolving *circulation networks* (CÂMARA et al., 2005).

4.1. Study area and its Transportation Network Dynamics

The Xingu-Iriri frontier is a region between two important rivers, the Xingu River and the Iriri River. It is a territory located in São Félix do Xingu and Altamira also known as Terra do Meio (Figure 4.1(a)). There are two main roads crossing the region, the Canopus road and the Farmers' road. Orthogonally to these main roads other secondary and auxiliary roads have been opened. In addition to that, there are a set of runways providing access entry points to the territory through small airplanes. All put together set up a multi-modal transportation network at a local scale named here as *circulation networks*. A schematic representation of it is shown in Figure 4.1 (b). It is with São Félix do Xingu that this territory establishes its strongest connectivity. The Canopus road was built in the end of 80´s. The Farmers' road was built in recent years (2002-2003). After its construction part of the Canopus road, near the Xingu River, was practically abandoned. Smallholders living nearby have undergone a decrease in their mobility and their access capabilities to reach for essential services and commerce.

Accordingly to Amaral et al (2006) the roads and rivers use patterns by the Terra do Meio population is seasonal. During the flood season, the get around is done mainly by the rivers. Some roads become blocked because of the large amount of holes, fall over bridges and roads segments entirely flooded. During the dry season, some of those roads start to be fixed and they come to be more used than the rivers. Runways offer a peculiar access entry point to the region. Actually, they operate as connection nodes in a multi-modal transportation network. Keeping in mind that in this area this multi-modal transportation network is based on precarious roads and rivers trails when available.

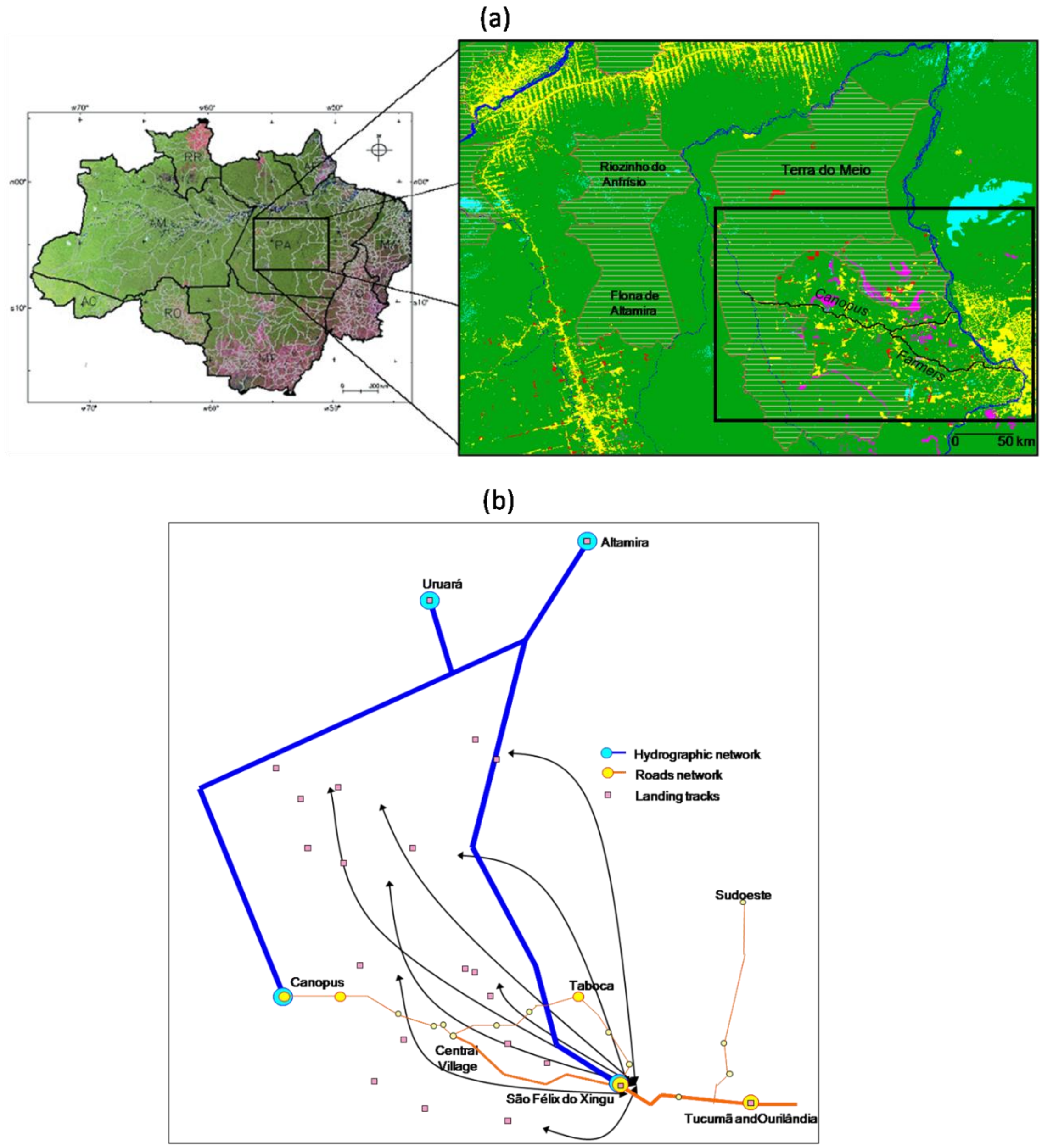

Figure 4.1 – (a) Legal Amazon, Terra do Meio and Xingu-Iriri Frontier (Rectangle area); (b) *Circulation networks* in the Xingu-Iriri frontier in 2006. [Source: adapted from (AMARAL et al., 2006)]

4.2. The Model

A Cellular-based Model is to be developed based on the process described in Amaral et al.(2006). The basis for the modeling is building an estimative of the potential proportion of deforested area in each cell. This estimative shall develop a function that can weight each cell in the Cellular Space considering its relative position to an entry point (connection node) of a *circulation network* that evolves over time.

In this model we must contemplate the evolution *circulation networks* by acknowledging new roads that are built, road segments that can no longer or temporarily be used, improvements on roads pavement and the opening and closing down of runways. In addition, the seasonality shall be dealt within the model, by altering the mobility and the connectivity patterns when the *circulation networks* are operating either in dry or in a flood season. The dynamics associated with this evolving *circulation networks* produce from a modeler's perspective the need for changing the neighborhoods structures including a re-evaluation of the its weights relations.

The proposed model will be calibrated using the 2000-2003 and 2003-2006 periods. It will make use of the PRODES data (PRODES) for establishing the deforestation patterns the running model shall be looking for. After the calibration, the 2006-2009 period will be simulated and the deforestation patterns resultant from that will be compared with the real data available.

5. SCHEDULER

To develop this work, we plan the following activities:

a) Expand and improve the presented concepts.
b) Implementation of the developed concepts in the TerraME platform.
c) Test the implemented structures.
d) Implementation of the model used as *proof-of-concept* experiment.
e) Elaborate a paper to submit to a conference (GeoINFO 2010).
f) Extend the paper to submit to a periodic.
g) Elaborate the thesis.
h) Defend the thesis.
i) Execute the corrections proposed by the examiners.

The following scheduler describes the when each activity will be executed:

Table 5.1 – Scheduler of the proposed work.

Activity	Mar/10	Apr/10	May/10	Jun/10	Jul/10	Aug/10	Sep/10	Oct/10	Nov/10	Dec/10	Jan/11	Feb/11	Mar/11
a	■	■	■										
b		■	■	■	■								
c		■	■	■	■								
d				■	■	■							
e						■	■						
f								■	■				
g								■	■	■	■	■	
h													■
i													■

6. CONCLUDING REMARKS

Several recent studies have demonstrated that the neighborhood configuration influences directly the outcomes of CA-based models. As neighborhoods change over time this becomes even more important for spatial dynamic modeling platforms. Modellers should be able to code their models in such a way that they would be able to cope with reconfiguration of its neighborhood structures at runtime.

At the end of this work, we hope the concepts developed here and its implementation on the **TerraME** platform will allow for representing the dynamics of an evolving *circulation network* that influences the patterns of deforestation in an Amazonian frontier site in the south of Pará. By demonstrating that, this dissertation could claim that the solution proposed extends the present TerraME modeling environment for dealing with a new class of problems, in which its dynamics implies neighborhoods structures that varies over time needing runtime reconfiguration to keep the model's hypothesis consistent over the model life cycle. We believe that the dynamic neighborhoods concept proposed here can be the basis for the implementation of this feature in several CA-based platforms for modeling and simulation of complex geographic phenomena.

7. REFERENCES

AGUIAR, A. P. D.; CÂMARA, G.; CARTAXO, R. **Modelling Spatial Relations by Generalized Proximity Matrices**. In: Proceedings of V Brazilian Symposium in Geoinformatics – GeoInfo 2003, Campos do Jordão, SP, Brazil, 2003.

ALMEIDA, C. M. **Modelagem da Dinâmica Espacial como uma Ferramenta Auxiliar ao Planejamento: Simulação de Mudanças de Uso da Terra em Áreas Urbanas para as cidades de Bauru e Piracicaba (SP)**. In: Remote Sensing Department, INPE, São José dos Campos, Brasil - Outubro, 2003a.

ALMEIDA, C. M.; BATTY, M.; MONTEIRO, A. M. V.; CAMARA, G.; SOARES-FILHO, B. S.; CERQUEIRA, G. C.; PENNACHIN, C. L. **Stochastic Cellular Automata Modelling of Urban Land Use Dynamics: Empirical Development and Estimation**. In: Computers, Environment and Urban Systems, Londres, Grã-Bretanha, v. 27, n. 5, p. 481-509, 2003b.

ALMEIDA, C. M.; MONTEIRO, A. M. V.; CAMARA, G.; SOARES-FILHO, B. S; CERQUEIRA, G. C.; PENNACHIN, C. L.; BATTY, M. **GIS and remote sensing as tools for the simulation of urban land-use change**. In: International Journal of Remote Sensing (Print), v. 26, p. 759-774, 2005.

ALMEIDA, C. M.; MONTEIRO, A. M. V.; CÂMARA, G. **Perspectivas Históricas de Modelos de Dinâmicas Urbanas e Regionais**. In: Geoinformação em urbanismo: cidade real X cidade virtual, Oficina de Textos (Ed), São Paulo, SP, Brazil, 2007.

ALVES, P. A.; AMARAL, S.; ESCADA, M. I. S.; MONTEIRO, A. M. V. **Explorando as relações entre a dinâmica demográfica, estrutura econômica e mudanças no uso e cobertura da terra no sul do Pará: lições para o distrito florestal sustentável da BR-163**. In: Geografia, Rio Claro (Printed), 2010. (in prelo)

AMARAL, S.; MONTEIRO, A. M. V.; CÂMARA, G.; ESCADA, M. I. S.; AGUIAR, A. P. D; **Redes e conectividades na estruturação da frente de ocupação do Xingu-Iriri – Pará (Network systems and connectivities structuring the colonization front in Amazonia: the case of Xingu-Iriri, in the South of Pará State, Brazil).** In: Geografia, Rio Claro, v. 31, n. 3, p. 655-675, September/December 2006.

AMARAL, S.; ESCADA, M. I. S.; ALVES, P. A.; ANDRADE-NETO, P. R.; PINHEIRO, T. F.; PINHO, C. M. D.; MEDEIROS, L. C. C.; SAITO, E. A.; RABELO, T. N. **Da canoa à rabeta: estrutura e conexão das comunidades ribeirinhas no Tapajós (PA)**. In: São José dos Campos, SP: INPE, 2009 (Research report).

ANDRADE-NETO, P. R.; MONTEIRO, A. M. V.; CÂMARA, G. **Entities and Relations for agent-based modelling of complex spatial systems**. In: Proceedings of I Brazilian Workshop on Social Simulation (BWSS/SBIA), p. 52-63, Salvador, BA, Brazil, October, 2008.

BAILEY, T.; GATTREL A. **Spatial Data Analysis by Example**. London, Longman, 1995.

BATTY, M.; XIE, Y. **From cells to cities**. In: Environment and Planning B, v. 21, p. 31–48, 1994.

BATTY, M. **GeoComputation using cellular automata**. In: Geocomputation, S. Openshaw and R.J. Abrahart (Eds), New York: Taylor & Francis, p. 95–126, 2000.

BATTY, M. **Cities and Complexity: Understanding Cities with Cellular Automata, Agent-based Models, and Fractals**. In: The MIT Press, Cambridge, Massachusetts, 2005.

BENENSON, I.; TORRENS, P. M. **Geosimulation: Automata-based modeling of urban phenomena**. Wiley, 2004.

BENENSON, I; KHARBASH, V. **Geographic Automata Systems: From the Paradigm to the Urban Modeling Software**. In: Proceedings of AGILE Conference, Estoril, Portugal, 2005.

BENENSON, I.; BIRFUR, S.; KHARBASH, V. **Geographic Automata Systems and the OBEUS Software for Their Implementation**. In: Juval Portugali (ed.), "Complex Artificial Environments Simulation, Cognition and VR in the Study and Planning of Cities", Berlin, Springer, 2006.

BOOCH, G. **Object-oriented analysis and design with applications**. Menlo Park, CA, Addison-Wesley, 1994.

CÂMARA, G.; AGUIAR, A. P. D.; ESCADA, M. I. S.; AMARAL, S.; CARNEIRO, T. G. S.; MONTEIRO, A. M. V.; ARAÚJO, R.; VIEIRA, I.; BECKER, B. **Amazon Deforestation Models**. In: Science, v. 307, p. 1043-1044, February 2005.

CÂMARA G.; VINHAS, L.; FERREIRA, K. R.; QUEIROZ, G. R.; SOUZA, R. C. M.; MONTEIRO A. M. V.; CARVALHO, M. T.; CASANOVA M. A.; FREITAS, U. M. **TerraLib: An open-source GIS library for large-scale environmental and sócio-economic applications**. In: B. Hall, M. Leahy (eds.), "Open Source Approaches in Spatial Data Handling". Berlin, Springer, 2008.

CARNEIRO, T. G. S. **Nested-CA: a foundation for multiscale modeling of land use and land cover change.** In: Computer Science Department – INPE, São José dos Campos, SP, 2006.

CARNEIRO, T. G. S.; CÂMARA, G.; MARETTO, R. V. M. **Irregular Cellular Spaces: supporting Realistic Spatial Dynamic Modelling over Geographical Databases**. In: Proceedings of X Brazilian Symposium on Geoinformatics – GeoInfo 2008, Rio de Janeiro, RJ, Brazil, 2008.

CLARKE, K. C.; HOPPEN, S.; GAYDOS, L. **A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area**. In: Environment and Planning B: Planning and Design, v. 24, issue 2, p. 247-261, 1997.

CLARKE, K. C.; GAYDOS, L. J. **Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore**. In: International Journal for Geographical Information Science, v. 12, p. 699-714, 1998.

COUCLELIS, H. **Cellular Worlds: A Framework for Modelling Micro-Macro Dynamics**. In: Environment and Planning A, v. 17, p. 585-596, 1985.

COUCLELIS, H. **Requirements for planning-relevant GIS: a spatial perspective**. In: Papers in Regional Science, v. 70, p. 9-19, 1991.

COUCLELIS, H. **From cellular automata to urban models: new principles for model development and implementation**. In: Environment and Planning B, v. 24, p. 165-174, 1997.

DEADMAN, P.; BROWN, R. D.; GIMBLETT, P. **Modelling rural residential settlement patterns with cellular automata**. In: Journal of Environment Management, v. 37, p. 147–160, 1993.

ENGELEN, G.; WHITE, R.; ULJEE, I.; DRAZAN, P. **Using Cellular Automata for Integrated Modelling of Socio-Environmental Systems**. In: Environmental Monitoring and Assessment, v. 34, p. 203-214, 1995.

ESCADA, M. I. S.; VIEIRA, I. C. G.; KAMPEL, S. A.; ARAUJO, R.; VEIGA, J. B.; AGUIAR, A. P. D; VEIGA, I.; OLIVEIRA, M.; PEREIRA, J. L. G.; CARNEIRO FILHO, A.; FEARNSIDE, P. M.; VENTURIERI, A.; THALÊS, M.; CARRIELLO F.; CARNEIRO, T. G. S. **Processo de ocupação nas novas fronteiras da Amazônia: o interflúvio do Xingu/Iriri**. In: Estudos Avançados, v. 19, p. 9-23, 2005.

FINKEL R. A.; BENTLEY, J. L. **Quad Trees a Data Structure for Retrieval on Composite Keys**. In: Acta Informatica, v. 4, n. 1, p. 1-9, March, 1974.

GIBSON, C. C. AND E. OSTROM, AND T. K. AHN, **The concept of scale and the human dimensions of global change: a survey**. In: Ecological Economics, (2000), 32(2), 217-239.

HAGOORT, M.; GEERTMAN, S; OTTENS, H. **Spatial externalities, neighbourhood rules and CA land-use modeling.** In: The Annals of Regional Science, v. 42, n. 1, March, 2008.

HENZINGER, T. A. **The Theory of Hybrid Automata**. In: Proceedings of the 11th Symposium on Logic in Computer Science (LICS'96), 1996.

HOWE, D. R. **Data analysis for data base design**. London, Ed-ward Arnold, 1983.

IERUSALIMSCHI, R.; FIGUEIREDO, L. H.; CELES, W. **Lua: an extensible extension language**. In: Software: Practice & Experience, v. 26, Issue 6, p. 635-652, 1996.

KOKABAS, V.; DRAGICEVIC, S. **Assessing cellular automata model behavior using a sensitivity analysis approach**. In: Computers, Environment and Urban Systems, v. 30, Issue 6, p. 921-953, November, 2006.

MOREIRA E. G.; AGUIAR A. P. D.; COSTA S. S.; CÂMARA G. **Spatial relations across scales in land change models**. In: X Brazilian Symposium on geoinformatics, GeoInfo 2008, Rio de Janeiro, RJ, Brazil, Novemver 2008.

MOREIRA, E. G.; COSTA, S. S.; AGUIAR, A. P. D.; CÂMARA, G.; CARNEIRO, T. G. S. **Dynamical coupling of multiscale land change models**. In: Landscape Ecology, v. 24, Issue 9, p. 1183-1194, November, 2009.

MORENO, N.; MÉNARD, A.; MARCEAU, D. J. **VecGCA: a vector-based geographic cellular automata model allowing geometric transformations of objects**. In: Environment and Planning B: Planning and Design, v. 35, p. 647-665, 2008.

MORENO, N.; WANG, F.; MARCEAU, D. J. **Implementation of a dynamic neighborhood in a land-use vector-based cellular automata**. In Computer, Environment and Urban Systems, v. 33, p. 44-54, 2009.

O'SULLIVAN D.; TORRENS, P. M. **Cellular models of urban systems**. In: Centre for Advanced Spatial Analysis, University College London, London (Paper 22), 2000.

O'SULLIVAN, D. **Graph-cellular automata: a generalized discrete urban and regional model**. In: Environment and Planning B: Planning and Design, v. 28, p. 687-705, 2001a.

O'SULLIVAN, D. **Exploring spatial process dynamics using irregular graph-based cellular automaton models**. In: Geographical Analysis, v. 33, p. 1-18, 2001b.

RODRIGUES, H. O.; SOARES-FILHO, B. S.; COSTA, W. L. S. **Dinamica EGO, uma plataforma para modelagem de sistemas ambientais**. In: Proceedings of XIII Simpósio Brasileiro de Sensoriamento Remoto, p. 3089-3096, Florianópolis, Brasil, 2007.

SOARES-FILHO, B. S.; CERQUEIRA, G. C.; PENNACHIN, C. L. **DINAMICA – a stochastic cellular automata model designed to simulate the landscape dynamics in Amazonian colonization frontier**. In: Ecological Modelling, v. 154, p. 217-235, 2002.

STRAATMAN, B.; HAGEN, A.; POWER, C; ENGELEN, G.; WHITE, R. **The Use of Cellular Automata for Spatial Modelling and Decision Support in Coastal Zones and Estuaria**. In: M. M. T. R. I. f. K a. Systems. Maastricht, The Netherlands: Maastricht University, 2001.

TAKEYAMA, M.; COUCLELIS, H. **Map Dynamics: Integrating Cellular Automata and GIS through Geo-Algebra**. In: International Journal of Geographical Information Systems, v. 11, Issue 6, p. 73-91, 1997.

TOBLER, W. **Cellular Geography**. In: Gale S. and Olsson G. (eds.), Philosophy in Geography, Reidel, Dordrecht, p. 379-386, 1979.

TORRENS, P., M.; BENENSON, I. **Geographic Automata Systems**. In: International Journal of Geographical Information Science, v. 19, n. 4, p. 385-412, April, 2005.

WAGNER, D. F. **Cellular automata and geographic information systems**. In: Environment and Planning B, v. 24, p. 219-234, 1997.

WHITE, R.; ENGELEN, G. **Cellular Automata and Fractal Urban Form: a Cellular Modelling Approach to the Evolution of Urban Land use Patterns**. In: Environment and Planning A, v. 25, p. 1175-1199, 1993.

WHITE, R; ENGELEN, G. **Cellular Dynamics and GIS: Modelling Spatial Complexity**. In: Geographical Systems, v. 1, p.237-253, 1994.

WHITE, R.; ENGELEN, G. **Cellular Automata as the Basis of Integrated Dynamic Regional Modelling**. In: Environment and Planning B, v. 24, p. 235-246, 1997.

ZHOU, B.; KOCKELMAN, K. M. **Neighborhood impacts on land use change: a multinomial logit model of spatial relationships**. In: The Annals of Regional Science, v. 42, n. 2, June, 2008.