Development of a Didactic Model of the Hydrologic Cycle Using the TerraME Graphical Interface for Modeling and Simulation

Tiago Lima[1], Sergio Faria[2], and Tiago Carneiro[1]

[1] TerraLAB, Computer Science Department, Federal University of Ouro Preto
Ouro Preto, Minas Gerais, Brazil
[2] Department of Cartography, Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
tiagofml@gmail.com, sergiofaria@ufmg.br, tiago@iceb.ufop.br

Abstract. Many modeling and simulation platforms provide general programming languages as interfaces for model construction. Some offers high-level modeling languages with conceptual basis and services to represent data structures and rules that will determine the model behavior. However, the direct use of a computational language is still a limiting factor to the broad usage of these platforms. Modelers often have different scientific backgrounds, presenting a lack of background on algorithms and programming techniques. Furthermore, there is no established methodology for model development. These problems confuse the modelers forcing them to deviate their attention from the problem being solved. We argue that a visual integrated development environment (IDE) can solve these problems, making easy the understanding and communication of the model conception and design. An IDE can also enforce the use of a common model development methodology. In this paper we describe a methodology for modeling Earth system phenomena using the TerraME GIMS tool, which is a visual IDE for the TerraME modeling and simulation platform. It enables users to build environmental models through visual metaphors that graphically describe models structure. We demonstrate the use of TerraME GIMS and present our methodology for the development of a didactic model for the hydrologic cycle. Future works include the development of diagrams to better describe the model behavior, including agent synchronization and communication.

Keywords: Environmental Modeling; Modeling Methodology; Visual Programming; TerraME GIMS; Integrated Development Environment.

1 Introduction

The Earth system comprises the interaction between socio-economic systems (of anthropic origins as the land use system) and biophysical systems (of natural origin as the ecological and atmospheric systems). In general, Earth system phenomena have a complex nature. This complexity requires the use of modeling and simulation techniques and tools to study, to understand and to represent systems behavior.

B. Murgante et al. (Eds.): ICCSA 2011, Part IV, LNCS 6785, pp. 75–90, 2011.

Furthermore, to deal with these problems it is necessary a multidisciplinary team of specialists from different fields of knowledge. One of the major challenges is to make explicit the differences that each team member has about phenomena behavior and about the model conception and design. Sophisticated user-friendly computational platforms are required to deal with this latter problem.

Modeling involves the construction of a simplified representation of the reality. It enables one to clearly define problems and concepts. It provides means of analyzing the observed behavior representing it in a synthetic environment and reporting simulation outcomes [1]. Modeling and computer simulation have been used in scientific researches to address problems of complex nature. When the solution has a high cost or cannot be obtained through experimentation [2]. Therefore, these methods and tools are essential to study terrestrial systems behavior. Typically, represented as spatial dynamic models, which describe spatial patterns of changes evolving over time [3].

The TerraME - Terra Modeling Environment [4] is a software platform for modeling and simulation of environmental phenomena which allows building dynamic spatial models integrated to Geographic Information Systems (GIS). Its main users are specialists involved in the development of environmental models and whose major knowledge is about the application domain, like geographers, ecologists, biologists, anthropologists, sociologists and economists. Although TerraME offers a high level programming language for model description called TerraML (TerraME Modeling Language), it still demands some programming skills from the user. During TerraME courses it is common the user to lose focus on modeling and TerraME concepts to pay attention on programming language syntax, style and on programming techniques. Therefore, a visual Integrated Development Environment (IDE) that allows model specification through graphical metaphors will be useful for keeping the focus of TerraME's users on solving problems in the application domain.

The TerraME GIMS (TerraME Graphical Interface for Modeling and Simulation) [5] is the graphical IDE of the TerraME platform. The TerraME GIMS is designed to make model development a more intuitive and effective task, increasing the productivity of the TerraME current users and decreasing the learning effort of new users. It enables users to visually describe models through interactions with graphical components such as menus, tool bars, model file trees, model structure specification diagrams, finite state machine diagrams, etc. The IDE automatically generates the TerraML source code corresponding to the model structure and behavior.

However, TerraME GIMS and TerraML modelers must understand the same basic concepts about the Nested-CA model of computation [4], which is the kind of automaton implemented by TerraME. This fact implies in the use of a common methodology for model development, which this paper briefly describes. To evaluate the use of a visual IDE for supporting this methodology, we have developed a didactic model to the hydrological cycle using both tools, the TerraML language and the TerraME GIMS interface. The hydrologic cycle model has been chosen due to the intuitive behavior of the processes it represents. Afterward, the manually generated and the automatically generated source code are compared to analyze their differences. This analysis demonstrates that the TerraME GIMS implements all

TerraME abstractions and can make easier and clearer the modeling process. Finally, the advantages and limitations on the use of TerraME GIMS to describe realistic environmental models for the TerraME platform are discussed.

2 TerraME and Nested-CA Basic Concepts

TerraME[1] is a software platform for building and simulating spatial dynamic computational models of geographical phenomena. It implements the Nested Cellular Automata (Nested-CA) model of computation, which provides essential features to study and represent Earth systems [4]. A realistic model construction is enabled through the TerraME integration with GIS, allowing to feed models with temporal series of detailed maps and sensors data.

TerraME users with good programming expertise can implement models directly using the TerraME C++ core. On the other hand, users with limited knowledge on programming can alternatively use the modeling language TerraML to build their models. This high level programming language is an extension of the LUA programming language [6] and adds special types and services for environmental modeling.

According to the Nested-CA conceptual design, which is in conformity with the Scale concept of Gibson et al. [7], the representation of a phenomenon in TerraME is performed from the description of the phenomenon behavior in time and space. The behavioral, spatial and temporal aspects of a phenomenon are viewed to form an indivisible model building block (or module) called Scale. Scales can be also seen as micro-worlds, i. e., as a synthetic (virtual) environment in which analytical entities (rules) change spatial properties over time. Scales can be nested to allow model decomposition, so that complex models (worlds) can be described from the composition of simpler ones. Each Scale can be used to represent different processes of a system, to model a different aspect of the same process, or to describe the same process at the different temporal or spatial resolution. Scales are implemented as Nested-CAs. This way, the Nested-CA design enables multiple scale model development.

In detail, the Nested-CA (or scale) is a container of two kinds of autonomous behavioral models, which may change spatial properties over time. The Automaton model is useful to represent continuous fields evolving over time, for example, temperature variation in a certain region. Its structure and functioning are based on the Cellular Automata Theory [8]. Meanwhile, the Agent model is based on the Agent Theory [9] and is useful to represent discrete entities or individuals, like institutions, persons, animals or vehicles. The internal state and the behavior of both models are determined by a Hybrid Automaton [10], whose structure and semantic allows the simulation of discrete, continuous or hybrid systems. Both models can communicate through message exchange. They also have access to a topological structure which defines their neighbors. The neighborhood structure is very flexible and is implemented as a weighted directed graph. The Nested-CA is also a container of cellular space [11] models, which are functions of indexes (like coordinates) into a set

[1] http://www.terrame.org/

of irregular cells used to represent the local aspects of the geographic space. Several discrete event schedulers can be embedded into a Nested-CA to describe the phenomenon dynamics based on the DEVS (Discrete Event System Specification) formalism proposed by Bernard Zeigler [12].

The TerraME software has been used in real case studies. Andrade et al. [13] developed models for spatial games and showed results demonstrating how mobility affects the Nash's equilibrium. A model for the simulation of fire propagation patterns in the Emas National Park, GO - Brazil was developed and presented by Almeida et al. [14]. Moreira et al. [15] used TerraME to understand how to represent spatial relation across scales in computational models for the land use system. TerraME has also been used by Pimenta et al. [16] in a case study in which it was analyzed how the existence of different rules of land use affects the landscape dynamics at regional level.

3 The TerraME Model Development Methodology

Modeling can be performed using different methods and processes [1,4,17]. However, we assume that models should be developed in an iterative and incremental way, such that at the end of each development cycle an improved model version is produced and the phenomenon becomes better understood. Each development cycle includes the following steps: (1) definition of the problem or scientific question, (2) development of the conceptual model, (3) choosing the appropriated modeling approach or paradigm, (4) development of the model by mathematical equations and relationships, (5) computational implementation including tests and code documentation, (6) calibration of model parameters, (7) evaluation of the model through comparison with empirical observations, (8) use of the model to explore alternative simulated scenarios, to make predictions and to support the decision making process.

During the first step, workshops, field works and literature review are essential activities. As the phenomenon behavior can be influenced by different driving forces at different scales [18], no single scale model can describe complex Earth System phenomena. Therefore, in the second step, the modeler should identify the major proximate factors and underlying driving forces affecting the phenomena behavior [19]. The modeler also needs to define the scales in which the phenomenon will be modeled. For this step, she has to choose the extents and resolutions those will be used to represent the phenomena in each scale dimension: behavior, space and time. The term extent stands for the magnitude of the measures used to represent the phenomenon. Resolution refers to the granularity of the measures [7]. Therefore, modelers have to decide about the actors and processes those will be taken into account. They should also choose the boundary of the region under study and the longest time period considered in the study. Afterward, the modelers should identify the resolutions in time and space in which changes in the observed behavior occurs. At this point, the modeler will have chosen the temporal, spatial and behavioral extents and resolutions in which the phenomenon will be modeled. Each chosen combination of extent and resolution for the dimensions space, time and behavior will give rise to one scale.

Nested-CA brings together the most useful abstractions to model environmental systems. The social aspects from Agent Theory, the spatial structures from the Cellular Automata Theory and the flexible specification of the dynamics allowed by the DEVS formalism are combined into a simple model of computation. The Hybrid Automaton used to implement Nested-CA Agents and Automata are capable of going through discrete changes expressed as internal transition rules, and of going through continuous changes described as differential equations. This way, in the third step the user can feel free to choose any of these paradigms to model different aspects of a phenomenon.

However, as Nested-CA Agents and Automata are autonomous machines, exogenous forces cannot ever change their internal state. Only internal transition rules and differential equations can make it. Hence, agents cannot give orders to other agents. But an agent can ask for instructions. The automaton in a central cell cannot directly change the internal state of automata within the neighbor cells. Nevertheless, it can change its internal state based on the information it gathered from its neighbors.

In the hierarchy of nested scales, scales in the higher levels must provide overall control to their internal scales. Each internal scale should be seen as a black box, i. e., as a module. For this reason, the information flow between scales should respect the hierarchy. Internal scales may change its state based on information coming from the parent scale. The parent scale must intermediate communication between internal scales. Elements in a scale should never directly change the internal state elements in other scales. Autonomy is the essential feature that legitimates the patterns that emerge in higher levels of organization from the local interactions between pair of Nested-CA components.

The modeler can use the TerraML or the TerraME GIMS tool in addition to the TerraView[2] GIS to accomplish the fourth and fifth steps. Finally, the TerraME VCToolkit can be used for model validation and calibration [20]. This way, modeling is a cyclic and incremental process, in which a model is built, reviewed and evaluated. In each cycle the understanding of the observed reality is expanded and improved.

4 TerraME Graphical Interface for Modeling and Simulation

As an extension of the Lua programming language, the TerraML syntax is too flexible to prevent modelers to produce source codes in which agents and automata directly change the internal state of each other. TerraME GIMS is a visual IDE for the TerraME platform. It enables the development of spatial dynamic models through the use of graphical representation of model structure and behavior. TerraME GIMS is implemented as a set of plug-ins that runs under the Eclipse[3] platform, adding capabilities to the visual specification and building of spatial dynamic models, automatically generating the model source code. TerraME GIMS has a layered software architecture, as illustrated in Fig. 1. It constitutes a new layer between end-users and the TerraME platform. The Eclipse platform appears as an intermediate layer between the TerraME modeling and simulation engine and the TerraME GIMS plug-in.

[2] http://www.dpi.inpe.br/terraview/

[3] http://www.eclipse.org/

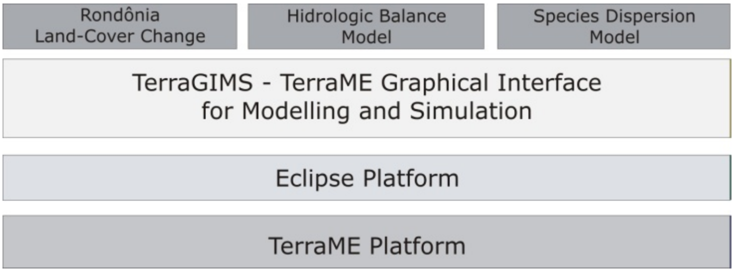
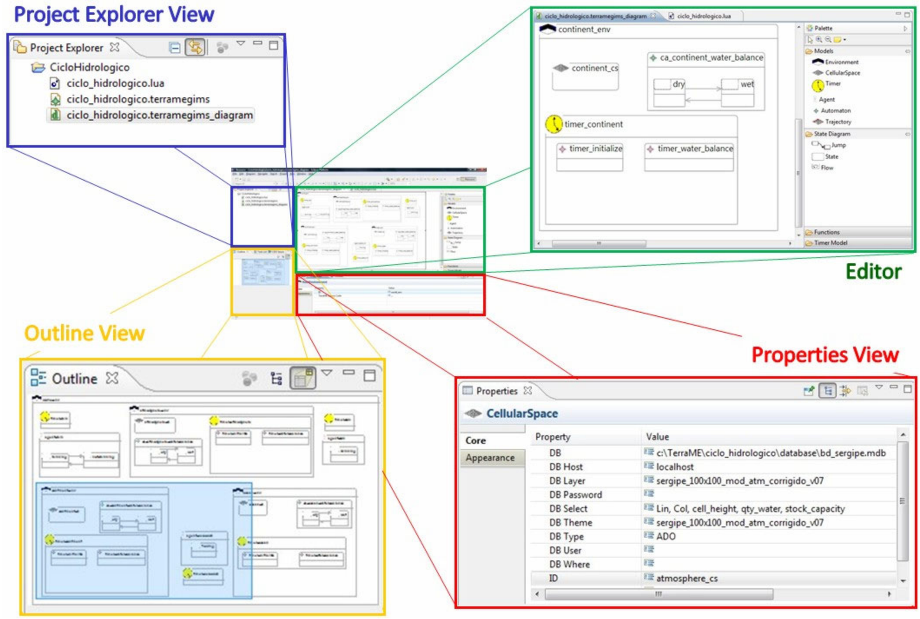

Rondônia Land-Cover Change
Hidrologic Balance Model
Species Dispersion Model
TerraGIMS - TerraME Graphical Interface for Modelling and Simulation
Eclipse Platform
TerraME Platform

Fig. 1. TerraME GIMS layered software architecture

The TerraME GIMS graphical user interface is composed by a main editor, called Graphical Editor, and the views Project Explorer, Outline and Properties, as illustrated in Fig. 2. Through Project Explorer View, users can access the files that are part of a project and navigate in model hierarchy. The hierarchy structure is shown as a tree of model components. The Graphical Editor shows the graphical representation of the model component selected in the Project Explorer. It is always possible to edit the component data structure and rules by changing its graphical representation. In Properties View, the values of model component properties can be visualized and edited through widgets as text fields, list boxes and check buttons. The Outline view provides a general overview of the model structure. This way, TerraME GIMS allows for the visual development of spatial dynamic models for Earth System phenomena.



Property	Value
DB	c:\TerraME\ciclo_hidrologico\database\bd_sergipe.mdb
DB Host	localhost
DB Layer	sergipe_100x100_mod_atm_corrigido_v07
DB Password	
DB Select	Lin, Col, cell_height, qty_water, stock_capacity
DB Theme	sergipe_100x100_mod_atm_corrigido_v07
DB Type	ADO
DB User	
DB Where	
ID	atmosphere_cs

Fig. 2. TerraME GIMS graphical user interface overview

5 Case Study: A Spatial Dynamic Model for the Hydrologic Cycle

A didactic spatial dynamic model for the Hydrologic Cycle was developed in order to evaluate the TerraME model development methodology and to assess the TerraME GIMS utility and correctness. The Hydrologic Cycle was chosen due to the intuitive behavior of its processes, making easier the interpretation of simulation outcomes. As this model requires the simulation of three different environments, i.e., land, ocean and atmosphere, it can be decomposed into three different nested scales. It also requires model integration with geographic database and the modeling of environmental phenomena from three different points of view: Spatial, temporal and behavioral. Furthermore, phenomenon behavior is simulated in the discrete and continuous way. Thus, this model is representative regarding the usage of data structures and services available to TerraME users. And it is also representative regarding the main challenges encountered in Earth System modeling.

In the following sections, the steps proposed in the TerraME model development methodology will be performed in order to build the hydrologic model: 1) problem definition, 2) conceptual model development, 3) computational and mathematical model definition, 4) geographical database and model implementation, 5) model calibration and validation, 6) simulation outcome analysis. A coastal region in the Brazilian state named Sergipe was chosen as study area.

5.1 Problem Definition: the Hydrologic Cycle

The hydrologic cycle (or water cycle) is the global phenomenon of water movement between the land surface and atmosphere. It can be considered a closed system at a global level [21, 22]. The hydrologic cycle occurs in the Earth's surface, in the

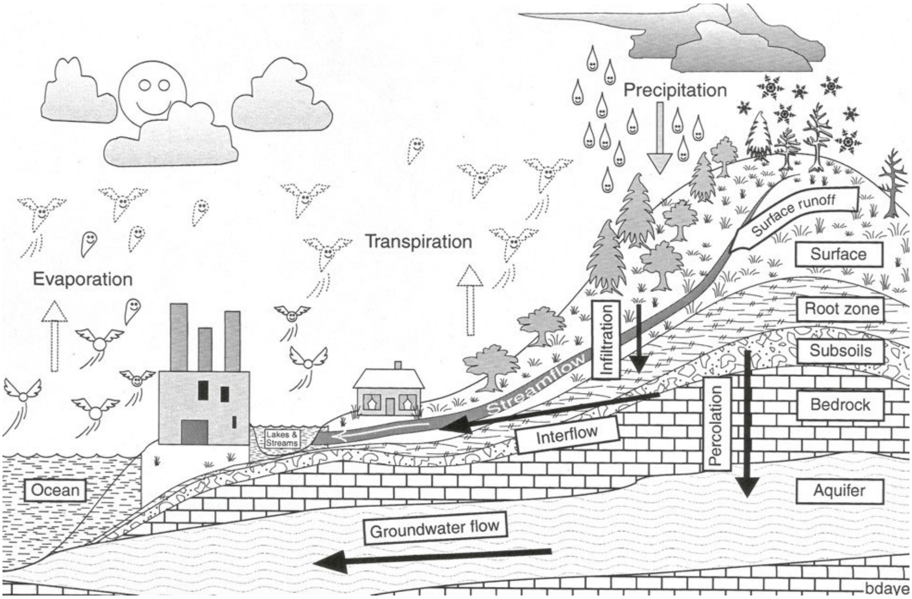


Fig. 3. Hydrologic cycle [21]

atmosphere, and in the interaction between them, as illustrated in Fig. 3. One phase of the cycle happens in land surface, which covers continents and oceans, where water circulation takes place inside and on the surface of soils and rocks, on oceans and on living beings. The water that circulates in the atmosphere comprises another phase. Finally, closing the cycle, there is an exchange between the water circulating on land surface and on atmosphere, which occurs in two ways: 1) towards of Earth's surface to atmosphere, where the transfer of water occurs primarily in vapor form, due to evaporation and transpiration phenomena; 2) towards of atmosphere to Earth's surface, where the transfer of water can happen in all the three physical states, being precipitation of rain and snow the most significant on a global scale [21].

5.2 Conceptual Modeling

In the hydrologic cycle, the interactions between atmosphere and Earth's surface (formed by the continents and oceans) happens from the water flow between them, which are represented by the evaporation (water flow from surface to atmosphere), precipitation (water flow from atmosphere to surface) and drainage (water flow from mainland to ocean) processes. For didactical reasons, the Earth's surface is separately modeled as two different micro-worlds or scales: Continent and ocean. Moreover, we assume that evaporation occurs only form the ocean to the atmosphere and that the precipitation occurs only from the atmosphere to the continent. In addition, the water flow also takes place inside the atmosphere, ocean and continent environments. The processes of convection, surface runoff and infiltration are also modeled. The water flow is mainly controlled by the region topography. On the continent surface and on the ocean bottom, the water flows from higher places to lower ones due to the surface runoff process. In the atmosphere, the water vapor flows from lower places to higher ones due to the convection process. The conceptual model is illustrated in Fig. 4.

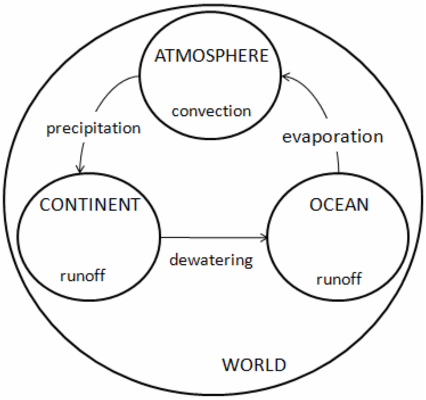


Fig. 4. Conceptual model of water flow in the hydrologic cycle

5.3 Computational and Mathematical Model Definition

In order to simplify the model development, the flows of water within the continent, ocean and atmosphere environments have been implemented in the same way. They have been represented by Cellular Automaton in which each cell is occupied the Hybrid Automaton (HA) shown in Fig. 5. A finite state machine with two states describes the HA discrete behavior. When in the state unsaturated, the water flow from higher cells to lowers cells is simulated by the exponential growth equation of the form $dW(t)/dt = k * W(t)$, where $dW(t)/dt$ is the instantaneous water flow, k is the flow coefficient in [0, 1] and $W(t)$ is the amount of water stored in the higher at time t. In the saturated discrete state, the HA continuous behavior simulates the distribution of water among lower neighbor cells, that is, lower neighbor cells receives the equal amount of water coming from a higher central. Therefore, this water flow is describe by the equation $dW(t)/dt = W(t)/N$, where N is the number of neighbor cells. The HA changes from the unsaturated state to the saturated state when the storage capacity where it is embedded is reached. In the opposite direction, when the stored amount of water is under the cell capacity, the HA goes back to the unsaturated discrete state.

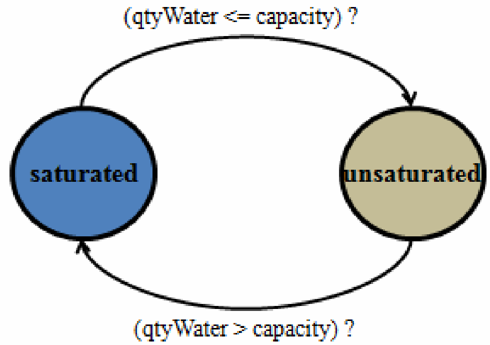


Fig. 5. Representation of the water flow hybrid automaton

The flows of water between scales, that is, the evaporation, precipitation and drainage processes have been implemented as Agents objects embedded in the "world" environment. These agents are also represented by the HA shown in Fig. 5.

5.4 Model Implementation Using TerraME GIMS

During the model geographic database construction, a SRTM image was used to represent the topography of the study area, a small size coastal region containing areas in the continent and ocean (Fig. 6). Then, the image was used to generate three regular grids of cells with the local attribute "altimetry" (Fig. 7). In TerraML, these grids of cells use to model space properties are named "CellularSpace".

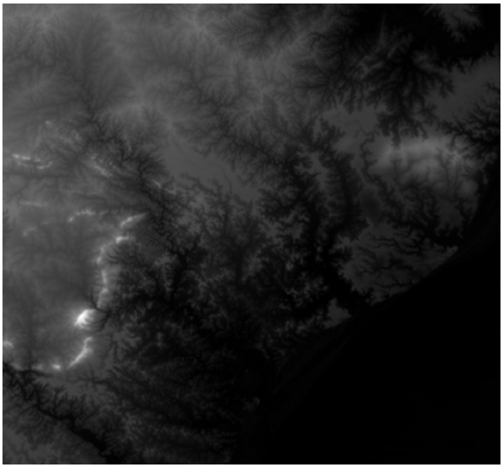


Fig. 6. SRTM image of the study area (SC–24–ZB). Source: Embrapa[4]

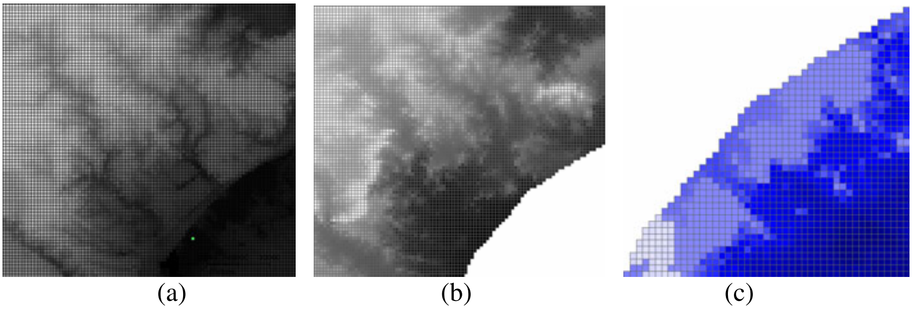
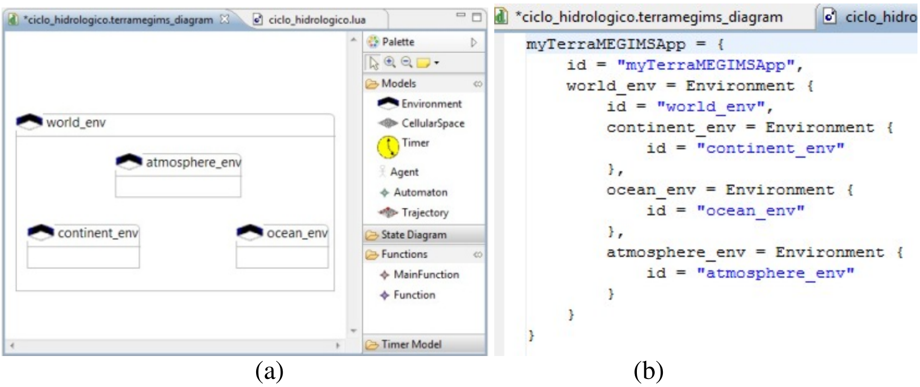


Fig. 7. Cellular spaces used to represent the study area topography in the three model environments: (a) atmosphere, (b) continent, and (c) ocean

In the TerraML, the Nested-CA concept of scale or micro-world is designated by the reserved word "Environment". In accordance with the Nested-CA structure, "Environment" objects are container of "Timer", "CellularSpace", "Automaton" and "Agent" objects. This way, the conceptual model was implemented as four nested Environment objects. The "world" environment represents the Earth and involves the environments "continent", "ocean" and "atmosphere" in its interior. For each one of these internal environments, the spatial and temporal behaviors of their inner processes were also implemented. A TerraME GIMS application is defined by an object named TerraMEGIMSApp located at the higher level of the model component hierarchy. This object is a container of "Function" and "Environment" objects. Using the TerraME GIMS graphical editor, one can visually specify the nested structure of the hydrologic cycle, as Fig. 8a shows. The automatically generated TerraML source code is shown in Fig. 8b.

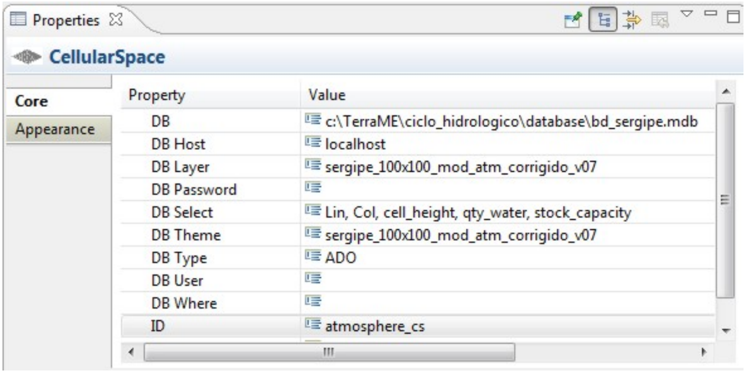
[4] http:// www.relevobr.cnpm.embrapa.br/



```
myTerraMEGIMSApp = {
    id = "myTerraMEGIMSApp",
    world_env = Environment {
        id = "world_env",
        continent_env = Environment {
            id = "continent_env"
        },
        ocean_env = Environment {
            id = "ocean_env"
        },
        atmosphere_env = Environment {
            id = "atmosphere_env"
        }
    }
}
```

Fig. 8. Definition of nested environments using TerraME GIMS: (a) graphical representation; (b) corresponding TerraML source code

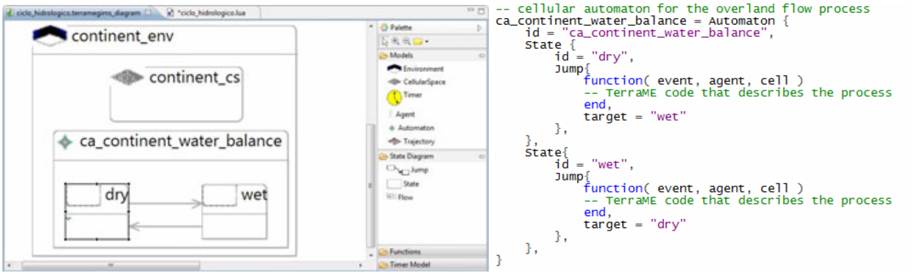
Then, the spatial model of each internal environment was implemented by the addition of CellularSpace objects to their interior. The properties of each CellularSpace object were edited directly in the TerraME GIMS Properties view, as illustrated in Fig. 9.



Property	Value
DB	c:\TerraME\ciclo_hidrologico\database\bd_sergipe.mdb
DB Host	localhost
DB Layer	sergipe_100x100_mod_atm_corrigido_v07
DB Password	
DB Select	Lin, Col, cell_height, qty_water, stock_capacity
DB Theme	sergipe_100x100_mod_atm_corrigido_v07
DB Type	ADO
DB User	
DB Where	
ID	atmosphere_cs

Fig. 9. Definition of spatial models

The behavioral model, shown in Fig. 5, has been implemented using the graphical metaphors presented in Fig. 10a. The finite state machine inside the continent environment has two states equivalent to the internal discrete states unsaturated (wet) and saturated (dry) discussed in section 5.3. The automaton's automatic generated TerraML code is also illustrated (Fig. 10b).



```
-- cellular automaton for the overland flow process
ca_continent_water_balance = Automaton {
    id = "ca_continent_water_balance",
    State {
        id = "dry",
        Jump{
            function( event, agent, cell )
            -- TerraME code that describes the process
            end,
            target = "wet"
        },
    },
    State{
        id = "wet",
        Jump{
            function( event, agent, cell )
            -- TerraME code that describes the process
            end,
            target = "dry"
        },
    },
}
```

(a) (b)

Fig. 10. The cellular automaton used to simulate the drainage process: (a) the graphical representation, and (b) the corresponding source code

Once defined the spatial and behavioral models, it remains only to define the temporal model. The temporal models for the environments continent, atmosphere and ocean are similar. They are composed by two pairs {event, message} (Fig. 11): (i) the first event occurs only in the model startup to initialize cell attributes, such as, waterQuantity and waterCapacity; (ii) the second event occurs at each simulation step to simulate the internal flows of water.

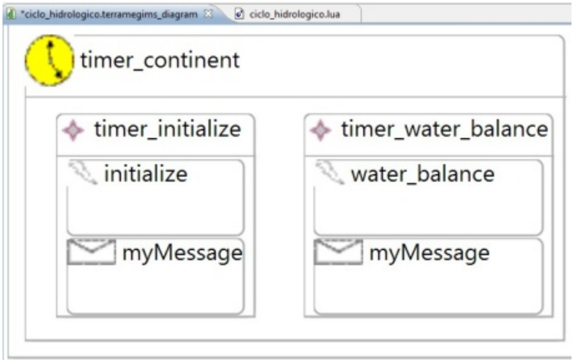


Fig. 11. Definition of continent temporal model

The flows of water between the environments occur through the processes of evaporation, precipitation and drainage and are implemented respectively by the agent agent_rain, agent_sun and agent_debouch. The agent agent_rain moves through the cellular space of atmosphere transferring part of water from each cell (in a saturated state) to the continent cell in the same geographical location. The agent agent_sun travels through the ocean cellular space transferring part of water from each cell to the corresponding cell in the atmosphere. The agent agent_debouch transfers part of water from each cell in coastal continent region to those cells in the ocean that are immediately adjacent to the continent cells. The temporal model of the environment "world" consists of three pairs {event, message} those execute the sun, rain and debouch agents. The graphical representation of hydrologic cycle model, including the spatial, temporal and behavioral dimensions, is presented in Fig. 12.

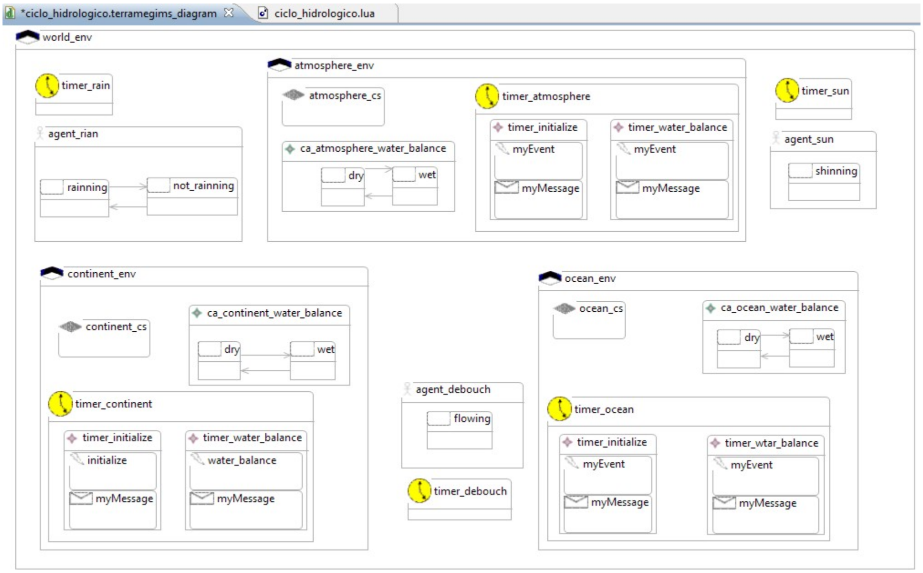


Fig. 12. Overview of the graphical representation of the hydrologic cycle model

5.5 Simulation Outcome Analysis

The model validation was done only to verify if the total quantity of water flowing in the model was keep constant (mass conservation) during all the simulation steps. The simulation outcomes are shown in Fig. 13. It is possible to observe the emergence of global patterns in the higher organization levels from the interaction of simple rules executed at the local levels.

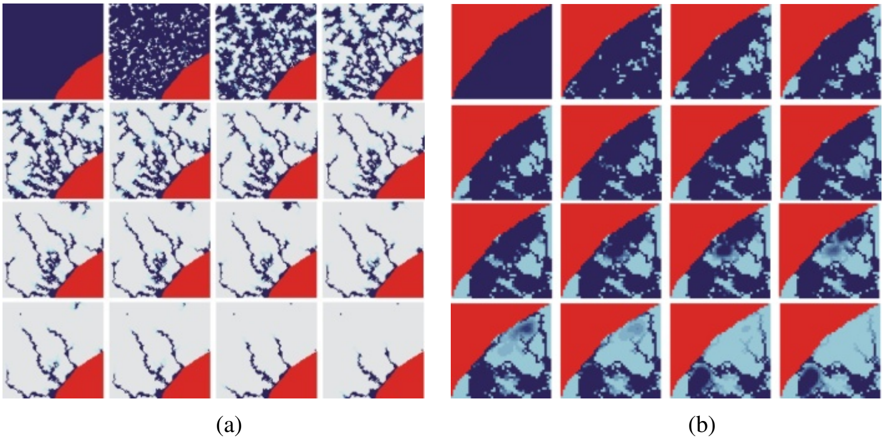
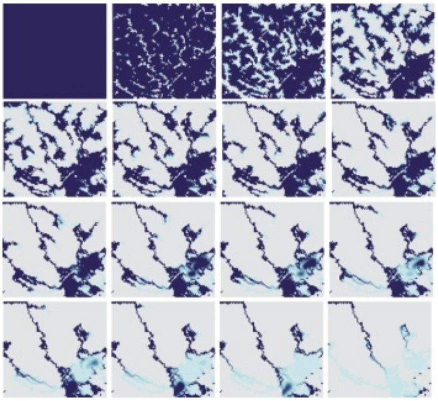


Fig. 13. The images showing the flow of water in each environment, from the left to right and from the top to the bottom: (a) water drainage in the continent, (b) water drainage in the ocean, (c) water vapor convection in the atmosphere. Legend: Red means “no data”, gray means “no water” and blue varies from dark to light to mean the amount of water in the cells.



(c)

Fig. 13. (*continued*)

6 Final Remarks

In order to evaluate the advantages brought by the use of a visual IDE in the development of spatial dynamic models of Earth System phenomena, we have outlined a methodology for model development using the TerraME modeling and simulation platform. Then, we have implemented an Eclipse plug-in, named the TerraME GIMS, which supports the use of this methodology. It allows graphical and interactive specification of spatial dynamic models whose source code is automatically generated. Finally, we have developed a spatial dynamic model to the hydrologic cycle using the TerraME GIMS plug-in. This didactic model illustrates the use of main functionalities of the TerraME platform and some of the challenges found in the development of multi scales models to Earth System phenomena.

One limitation in the use of graphical interface for modeling is the lack of flexibility regarding the customization of the automatically generated code. On one hand, the standardized structure of the generated code may limit users to freely organize their model. On other hand, it can be a great benefit to users which do not have much experience in programming and code organization, contributing to the model readability and maintainability. Moreover, another advantage of automatic code generation is the reduction of the user's effort to understand and use the rigid syntax of programming languages, allowing them to focus on the phenomenon representation.

Moreover, the communication of the model conception and design through a graphical representation allows modelers to identify the model components and the relationships between them in an easier and more intuitive way. The graphical representation to the hydrologic cycle model is very close to the conceptual model shown in section 5.2.

Although TerraME GIMS is still a prototype, it is capable of graphically represent Nested-CA basic concepts. It can be use to completely describe model structure. However, some effort is still required to make possible the complete description of the model behavior through graphical metaphors. Future works also include the evaluation of the TerraME GIMS in experiments involving groups of users. Measures such as modeling speed, frequency of errors, effectiveness of model communication still have to be collected. Both TerraME and TerraME GIMS are under continuous development at TerraLAB[5] [23].

Acknowledgements. We would like to thank CAPES and CNPq (CTINFO-2010) for the financial support.

References

1. Turner, M.G., Gardner, R.H., O'Neill, R.V.: Landscape Ecology in Theory and Practice: pattern and process. Springer, New York (2001)
2. Bratley, P., Fox, B.L., Schrage, L.E.: A guide to Simulation, 2nd edn. Springer, Heidelberg (1987)
3. Pedrosa, B.M., Câmara, G.: Modelagem dinâmica e sistemas de informações geográficas. In: Meirelles, M.S.P., Câmara, G., Almeida, C.M. (org.) Geomática - Modelos e aplicações ambientais, Embrapa, Brasília, DF, Brasil, vol. 5, pp. 235–280 (2007)
4. Carneiro, T.G.S.: Nested-CA: a foundation for multiscale modeling of land use and land change. Ph.D. thesis, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, Brasil (2006)
5. Lima, T.F.M., Carneiro, T.G.S., Faria, S.D.: Desenvolvimento de uma plataforma gráfica para a descrição de modelos de sistemas ambientais. In: Proceedings of the X Brazilian Symposium on GeoInformatics, pp. 121–126. Sociedade Brasileira de Computação (SBC), Brasil (2008)
6. Ierusalimschy, R., de Figueiredo, L.H., Filho, W.C.: Lua - an extensible extension language. Software Practice and Experience 26(6), 635–652 (1996)
7. Gibson, C.C., Ostrom, E., Ahn, T.K.: The concept of scale and the human dimensions of global change: a survey. Ecological Economics 32(2), 217–239 (2000)
8. Neumann, J.V.: Theory of self-reproducing automata. Mathematics of Computation 21(100) (1996)
9. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–152 (1995)
10. Henzinger, T.A.: The theory of hybrid automata. In: Proceedings 11th Annual IEEE Symposium on Logic in Computer Science, pp. 278–292. IEEE Computer Society Press, Los Alamitos (1996)
11. Carneiro, T.G.S., Câmara, G., Maretto, R.V.: Irregular cellular spaces: Supporting realistic spatial dynamic modeling over geographical databases. In: Proceedings of the X Brazilian Symposium on GeoInformatics, pp. 109–120. Sociedade Brasileira de Computação (SBC), Brasil (2008)
12. Zeigler, B.P.: Theory of Modeling and Simulation, 1st edn. John Wiley, Chichester (1996)
13. Andrade, P.R., Monteiro, A.M.V., Câmara, G., Sandri, S.: Games on cellular spaces: How mobility affects equilibrium. Journal of Artificial Societies and Social Simulation 12(1), 5 (2009)
14. Almeida, R.M., Macau, E.E.N., França, H., Ramos, F.M., Carneiro, T.G.S.: Simulando padrões de incêndios no parque nacional das Emas, estado de Goiás, In: Proceedings of the X Brazilian Symposium on GeoInformatics, pp. 183–194. Sociedade Brasileira de Computação (SBC), Brasil (2008)

[5] http://www.terralab.ufop.br/

15. Moreira, E.G., de Aguiar, A.P.D., Costa, S.S., Câmara, G.: Spatial relations across scales in land change models. In: Proceedings of the X Brazilian Symposium on GeoInformatics, pp. 95–107. Sociedade Brasileira de Computação (SBC), Brasil (2008)
16. Pimenta, P.F., Coelho, A., Costa, S.S., Moreira, E.G., Aguiar, A.P., Câmara, G., Araújo, R., Ribeiro, A.: Land change modeling and institutional factors: heterogeneous rules of territory use in the Brazilian Amazonia. In: Proceedings of X Brazilian Symposium on GeoInformatics, pp. 81–93. Sociedade Brasileira de Computação (SBC), Brasil (2008)
17. Hannon, B.M., Ruth, M.: Dynamic Modeling (Modeling Dynamic Systems), 2nd edn. Springer, New York (2001)
18. Aguiar, A.P.D., Câmara, G., Escada, M.I.S.: Spatial statistical analysis of land-use determinants in the Brazilian Amazonia: Exploring intra-regional heterogeneity. Ecological Modelling 209(2-4), 169–188 (2007)
19. Geist, H.J., Lambin, E.F.: Proximate causes and underlying driving forces of tropical deforestation. BioScience 52(2), 143–150 (2002)
20. Fraga, L.M., Lana, R.M., Carneiro, T.G.S., Guimarães, F.G.: Calibração em modelagem ambiental na plataforma TerraME usando algoritmos genéticos. In: Proceedings of LXII Brazilian Symposium on Operational Research (2010)
21. da Silveira, A.L.L.: Ciclo Hidrológico e Bacia Hidrográfica. In: Tucci, C.E.M. (org.) Hidrologia: Ciência e Aplicação, ch. 2, pp. 35–51. UFRGS, Porto Alegre, RS, Brasil, 3 (2003)
22. Ward, A.D., Trimble, S.W.: Environmental Hydrology, 2nd edn. CRC Press, Boca Raton (2003)
23. Carneiro, T.G.S., de Lima, T.F.M., Faria, S.D.: TerraLAB - Using free software for Earth system research and free software development. In: Proceedings of the X Workshop of Free Software, pp. 35–40. Sociedade Brasileira de Computação (SBC),Porto Alegre, RS, Brasil (2009)