Capturing Spatial Simulation specifics in Geographic Information Systems with a UML Profile

Luís Moreira de Sousa[1] and Alberto Rodrigues da Silva[1]

Instituto Superior Técnico, Lisboa, Portugal
luis.moreira.de.sousa@ist.pt alberto.silva@acm.org

Abstract. This article describes a model-driven domain specific language for Spatial Simulation in the field of Geographic Information Systems (GIS) called DSL3S. Techniques such as cellular automata and agent-based modelling have long been used in this field to capture and simulate the dynamics of change of spatial information. The tools available to apply these techniques either require programming skills or impose serious constraints on application scope, which has limited their usage among users. DSL3S aims at synthesizing the relevant concepts of spatial simulation in the GIS context through a UML profile. This profile shall be used to develop simulation models by graphical composition of elements like classes and relationships, from which code generation mechanisms can produce ready (or near ready) to run simulations.

Keywords: spatial simulation, domain specific language, UML profile

1 Introduction

Stakeholders of an Information System need not only to know how the data changed in the past, but in order to plan ahead or otherwise reason upon the data, they need also to understand why it changed the way it did and how it might continue to evolve into the future. In the GIS realm in particular, this need is even more prominent due to the underlying time dependence of spatial data. This need is met recurring to two processes that are part of the same technology: Spatial Modelling and Spatial Simulation. Modelling is the process by which the fundamental drivers of change - the Spatial Dynamics - are captured into mathematical, logic or functional constructs. Simulation is a process through which a model is applied to a set of data during a certain period of time [7].

The oldest od the tools used for Spatial Simulation are Cellular Automata (CA) [19] in which the world is discretized in a grid of equal sized cells, evolving in accordance to a fixed set of rules. More recently, Agent-based Simulations have become a popular tool that has also been applied to spatial information systems. An agent can be defined as an autonomous object that perceive and react to its environment [17]. Agent-based Simulations and CA are two concepts that superimpose to some extent in the GIS context, but the former brought new possibilities, for agents can also store knowledge, reason before acting and even

learn. Beyond that, agents can be used to model phenomena that do not have exact geographic meaning, such as social or economic interactions.

On the GIS related sciences, Spatial Simulations have been used extensively. The following application domains can be highlighted: (i) *Urban Planning* - understanding and forecasting changes in urban extents [1]; (ii) *Land Use* - studying the dynamics of land use, e.g. changes between agriculture and forest [11]; (iii) *Forestry/Wild Fire* - studyding forest growth and anticipating fire spread [8]; (iv) *Biology* - modelling habitat evolution and population dynamics [4].

Presently a spectrum of Spatial Simulation tools can be devised, ranging from those that present support at Program-level, closer to the programming language, to those that operate at Model-level, closer to the conceptual model that represents reality [5]. In the middle of this spectrum lay Domain Specific Languages (DSL).

Program-level support tools extend the facilities available in general-purpose programming languages, usually as code libraries. This approach substantially reduces coding time and can increase program reliability. Higher-level code, usually in a general-purpose OO programming language, specifies how objects are used to produce the desired model behaviour. They require strong programming skills from the user and impose a relevant learning curve to their mastering. Inherently open-source, they tend to gather large communities of users and usually provide a good level of GIS interoperability. Examples of these tools are Swarm [12], RePAST [2] and MASON [9].

Model-level support tools allow the usage of spatial simulation models without requiring programming. These are pre-programmed models, designed for specific application fields that can be parametrized by the user. They allow faster model development and provide fairly straightforward mechanisms for implementation, though invariably constraint the modeller to a specific application and dynamics framework. They tend to be commercial tools, taking advantage of market niches; interoperability with spatial data is usually weaker than with Program-level tools, sometimes reliant on third party software. Examples include: TELSA [10], LANDIS [13] and SLEUTH [20].

Midway between Program-level and Model-level support tools are domain specific tools, usually providing Model-level support for a range of application domains. They make fewer assumptions about the underlying model, often providing ways of developing new behaviours, or at least refining them. Model coding is performed in a restricted environment where behaviour is described using simple constructs, encapsulating traditional coding activities. Examples are NetLogo [15], SELES [5] and MOBYDIC [6]. Existing efforts for spatial simulation DSL have so far remained on declarative or functional languages. Beyond that, interoperability with spatial data has remained an area with week support by these tools.

This article describes the DSL3S project, an effort to bring Spatial Simulation closer to GIS end users with a model-driven approach; a tool requiring little to none programming skills without restricting application domain or compromising interoperability with spatial data. In Sec. 2 is presented the UML profile

that underpins the project and in Sec. 3 its application to two popular simulation models. Sec. 4 sums up the article and discusses future work.

2 UML Profile Prototype

The UML 2.0 specification allows the extension of its core primitives (graphical elements, links, etc) by specialization for particular domains [14]. This is achieved with the definition of a UML Profile, a collection of stereotypes, tagged values and constraints. Stereotypes are specializations of existing UML model elements, defining new elements representing a narrower abstraction. A semantically related set of stereotypes, specified by tagged values and restrictions, can thus be used to customize the UML into a new specialized language dedicated to a certain domain.

The Domain Specific Language for Spatial Simulation Scenarios (DSL3S) is a UML profile that defines a set of stereotypes enclosing the abstract notions behind spatial simulation. These stereotypes can be seen as the terms used when explaining a simulation in natural speech, such as describing a fire as an agent, or slope as a spatial variable. A successful UML profile will allow the development of simulation models by applying these stereotypes to classes and creating the correct relations between them. Once the model is built it can then be parametrized by toggling the values of its tagged definitions.

This DSL3S takes spatial simulation as a branch of the wider Spatial Analysis GIS field, where models' inputs originate at least partially from a GIS and whose outputs may also have geo-referenced relevance, in which case can be returned to a GIS. As of now, only non-adaptive agents are considered, and all agents are assumed to exist in the space of simulation, thus forcefully being geographic entities. One and the same approach shall be taken to both models that are traditionally implemented with cellular automata and those based on agents, seeking a language abstracted from such technical differences.

2.1 The Meta-model

Three main pillars have been identified as the underpinning concepts of a spatial simulation: Spatial Environment variables, Animats and Behaviour. **Spatial Environment** variables are spatial information layers that have some sort of impact on the dynamics of a simulation, e.g. slope that deters urban sprawl, biomass that feeds a wildfire. **Animat** is a term coined by S. W. Wilson [18] signifying *artificial animal*; here it is used more widely, representing all types of agents of change such as wildfires, urban areas or agents in a predator-prey model. **Behaviour** associates the former two concepts, defining how Animats react to their surrounding environment and internal state; examples can be movement or replication.

Beyond these basic concepts, other elements can also be found in a simulation. There may exist **Global** variables, setting information that is constant across the whole space of simulation, such as wind direction in a wildfire model.

Another important sort of context variables are those that support Animat internal state. The Animat's **State** can be seen as a set of variables that describe each instance at a certain moment in time.

Figure 1 presents these elements in a conceptual model. A **Simulation** is composed by a set of Spatial Environment variables, Animats and Global variables; Animats may be composed by series of State variables. Animats are also composed by Behaviours that determine how it's internal state evolves; behaviours can be function of global variables, spatial environment or the state of other animats.

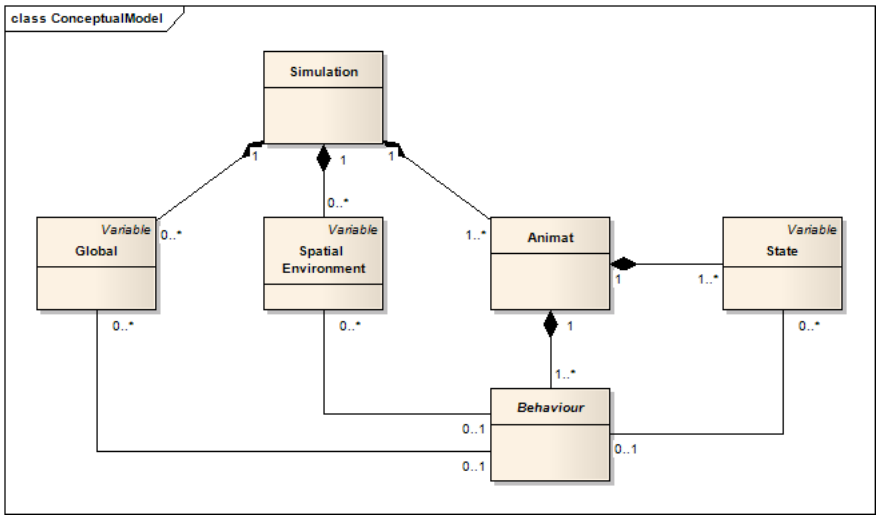

Fig. 1. The Profile's core concepts.

All these concepts are stereotypes in the DSL3S profile. A stereotype named Simulation will be used to capture definitions such as the number of time steps to run, the simulation's spatial extent and type of spatial neighbourhood. It may also be used to parametrize other features such as data display or data logging mechanisms. Figure 2 shows these main stereotypes together.

The Global variable stereotype is intended to be a simple scalar value that may vary with time. It can for instance be set randomly at simulation start and/or made to vary randomly each time set; it can also be fed into the model as a pre defined time-series of values, that, for instance, may be imputed from a text file.

The stereotype for Spatial Environment is essentially a stub for the input of geo-referenced data. Each instance shall correspond to a spatial layer (either in raster or vector format) with the characteristic of having an unequivocal value for each location in space. This stereotype shall also provide means for the fully

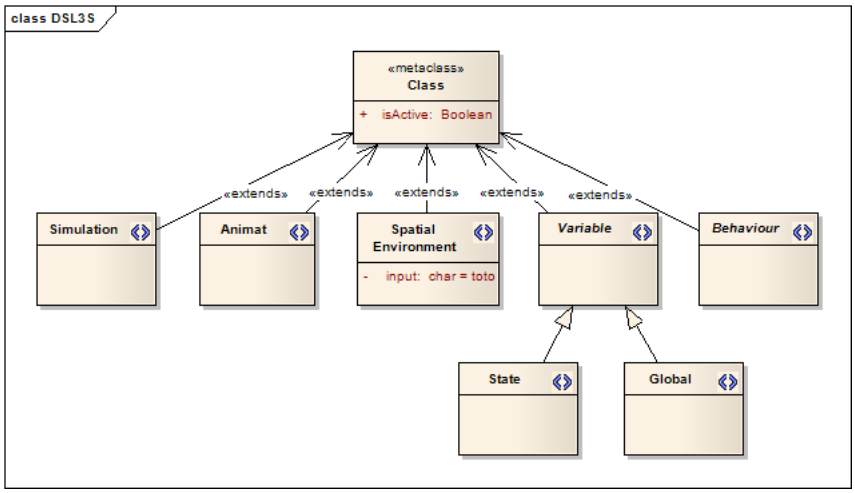

Fig. 2. The Profile's main stereotypes.

random generation of spatial environment variables, that may be useful for the creation of synthetic scenarios.

Animats are essentially an aggregation of state variables residing at a perfectly identifiable location in space. Each instance of this stereotype models a specific genre of animat (e.g. wolves and sheep in a predator-prey simulation). The initial number of animats instances and their spatial positioning can be provided by a specific geo-referenced data set such as a raster map. By similar mechanisms the initial values of state variables can too be set with geo-referenced data. These initial settings can also be randomly generated for simulations where it may apply.

The elements considered so far focused on the information needed to run a spatial simulation. But more than that is required to capture spatial dynamics, the way animats act has to be made explicit. The option has been to model this aspect with the following specializations of the Behaviour stereotype:

- **Initiate** - captures the conditions under which a new instance of an animat can appear in the simulation, the act of "birth". An example may be an urban development simulation where the emergence of new urban spots is possible in an area that meets a certain set of criteria, like distance to transport infrastructure or topography. Tagged values of this class will allow the setting of neighbourhood thresholds relative to spatial environment variables, or other animats, above which the "birth" of a new animat becomes possible. Probabilities of birth will also be parametrizable with tagged values in this class. The initial internal state of a new animat will be parametrizable also with tagged values, but in the Animat class itself.
- **Move** - relates an animat with one or more spatial environment variables or other animats determining the locations that are more or less favourable

to be in. This class will provide tagged values that allow to weight the relevance of each related class. For instance, in a predator-prey simulation the movement of a "sheep" animat class may be positively weighted in a relation with a "grass" class and negatively weighted in a relation with a "wolf" animat class.

- **Replicate** - captures behaviours where an animat replicates itself to an adjacent location, such as a wildfire spreading or an urban area sprawling. Just as with previous behaviours, the objective is to capture the conditions under which an animat may originate a sibling into its neighbourhood. Tagged values may weight the influence of spatial environment variables (e.g. fire spread) or set thresholds against the animat's internal state (e.g. biological reproduction). Impact on the reproducing animat's internal state will also be modelled with specific tagged values. As with the Initiate behaviour, initial internal state of a new animat is parametrized in the animat's class itself.
- **Harvest** - an act on which an animat may change other elements in the same spatial location; it can act on an environment variable, such as a wildfire consuming bush, or by seizing another animat as in a predator-prey simulation. In this class tagged values will exist essentially to parametrize the changes of both harvester and harvested. This is modelled with an harvest rate (e.g. the rate by which a fire consumes bush), and the impact on the harvester animat's internal state. A consumption rate of 100% may be used to model preying relationships.
- **Perish** - defines the circumstances under which an animat instance may cease to exist during simulation (e.g. "starve"). This class will in most cases define minimum thresholds related to the animat's internal state that determine conditions for the endurance of its existence. This class may be associated with the animat class itself to set conditions by which an instance may cease to exist due to crowding. Associations with spatial environment variables may provide further conditions for the existence of an animat.

This is just a selection of behaviour classes that provide a set of core procedures for an animat's conduct. In their seminal book, Epstein and Axtel [3] conceive a considerably larger set of behaviours, including elaborate processes such as trade and cultural exchange. While interesting, these intricate behaviours are not common in pure GIS applications of agent based simulation (where the dynamics is captured at a higher level of geographic abstraction) and more keen to Social or Economics studies. However, these behaviours can be added in the future to the profile if necessary, the DSL3S being an open language. The addition of further behaviours will in fact be the main process of extension of the language.

2.2 Proposed Views and Icons

Models built with the DSL3S can become rather complex visually if a single diagram is used to represent all classes, tagged values and associations. To avoid

such difficulties and provide a through framework for the development and presentation of models with the DSL3S, a scheme of Views is here proposed. These views aim at displacing the model in such a way that each aspect of a simulation is presented in a specific diagram.

The first view proposed is the **Simulation View** which is simply a package diagram that aggregates all other views. Each package in this view holds a diagram that describes a particular aspect of the simulation. From the Simulation View branch three others: Global View, Spatial View and Animat View.

To model the simulation's settings that do not have spatial realization there is the **Global View**. Here is parametrized the number of time steps to run, spatial extent or result output. Here are also set global variables that may influence the simulation. In this view only the stereotypes Simulation and Global variable are applicable; the Simulation stereotype should be applied to only one class.

Containing the simulation's geographic inputs is the **Spatial View**, where all the Spatial Environment variables are configured. In this View only the Spatial Environment stereotype is used, presenting in a single diagram all the spatial layers that are inputs to the model. Classes in this view must be linked to the Simulation class defined in the Global View with composition relationships.

The actors of change are modelled in the **Animat View**, where the properties defining animat internal state are set. In this view only the Animat and State stereotypes apply, with composition relationships linking animats to the main Simulation class. In simulations where more than one animat may exist several animat views can be used, one for each animat.

For models that are not composed by a large number of variables, an **Integrated View** can be used to present all the model's inputs together in a single diagram. In this view all the elements of the Global, Spatial and Animat views can be presented together, with their respective relationships to the single Simulation class.

Lastly there is the **Behaviour View**, where each behaviour and its relationships to other classes are configured. This view may contain classes only with the stereotype Behaviour applied on, but in the diagram will also be represented the composition links to the respective animat and all other relationships with other classes that parametrize each behaviour. To improve readability and model organization it is also proposed the creation of dedicated views for each different behaviour in the model. This way behaviour views can be named as Behaviour.Move, Behaviour.Replicate, etc. Figure 3 presents the Views scheme proposed as a package diagram.

Beyond these views a set of icons is also proposed to make the language visually explicit. For each of the stereotypes is presented an image pretending to capture basic semantics of a simulation. For the stereotypes Simulation, Global and Spatial Environment are used direct pictorial representations of their concepts. For the stereotypes Animat and State are used abstract symbols intending to create mental associations with a simulation model. As for behaviour stereotypes, the icons proposed use the abstract animat symbol with graphical

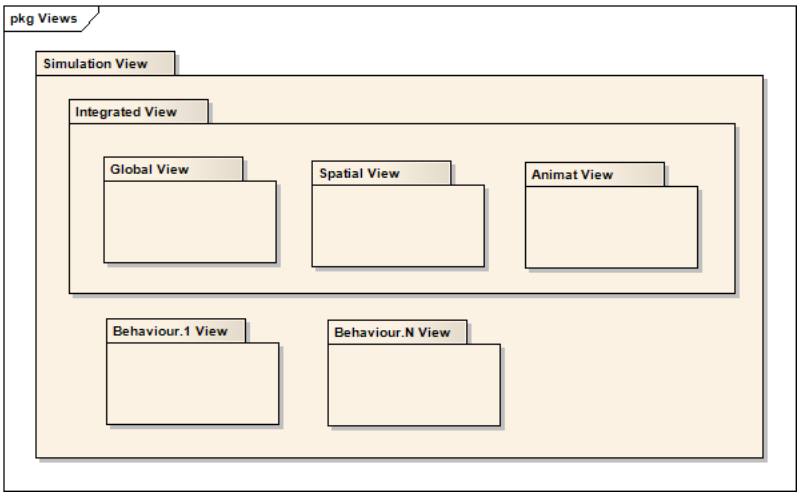

Fig. 3. The proposed Views scheme for DSL3S.

allusions to the actions of a small animal or bug. Figure 4 presents again the conceptual model of DSL3S, this time with the stereotype icons.

3 Application example

Although the DSL3S may be considered everything but complete at this stage, it seems to enclose enough concepts to describe a traditional simulation in a GIS context. In this section a typical simulation is described using DSL3S's stereotypes and views, showcasing its usage.

A popular application of spatial simulation techniques has been in Urban Development, where the model-level tool SLEUTH has acquired notoriety by it's successful application to different parts of the World [16,20]. There's a strict set of inputs to the program, of which five are spatial data sets: (i) Slope - a deterrent of urban sprawl; (ii) Exclusions - areas where cities can't develop into; (iii) Land Use - a thematic map classifying terrain occupation; (iv) Transportation - roads, rail lines, waterways, etc; and (v) Urban Areas - areas already developed at the beginning of simulation.

The first stereotype to apply in the model is Simulation. A tag termed *steps* can here be used to parametrize the number of time steps to run in each simulation; this is a mandatory parametrization, that is set with a default number when the stereotype is applied. Also in the Simulation class can be defined the spatial extent of simulation, by the parametrization of four bound tags; if not provided, the simulation shall proceed within the intersection of all spatial variables and animats inputs extents.

Next the spatial variables, all the data sets described above could be considered as spatial environment variables, but the Urban Areas dataset is in fact

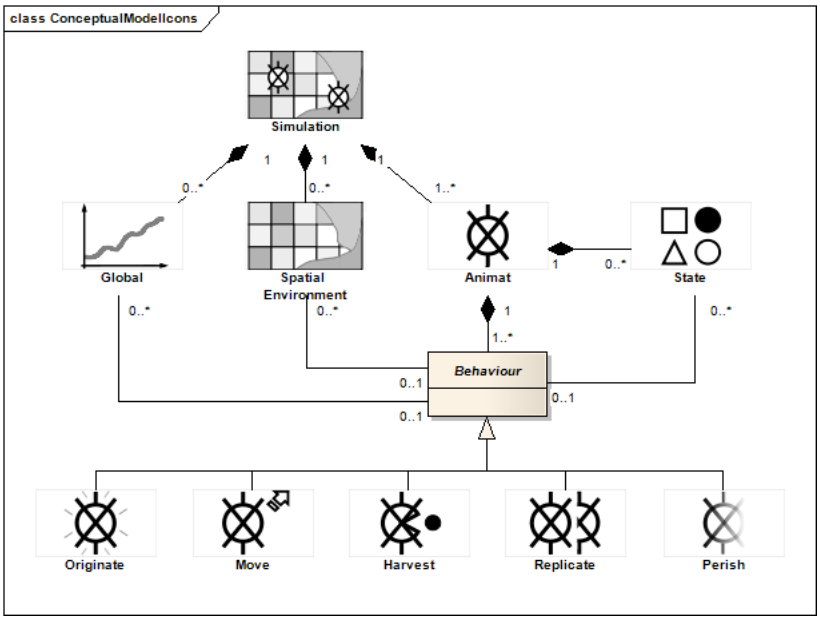

Fig. 4. The proposed Icons for DSL3S.

an animat class, since it behaves in ways that increase its extent. Put another way, each urban cell in the dataset can be considered an animat that replicates itself throughout simulation. Spatial environment stereotypes are parametrizable with two essential tags: *input* and *ouput*, which set file system paths to input and output files. The parametrization of the *input* tag is mandatory, without which the simulation won't run. The *output* tag provides the path to where the dataset should be saved at the end of simulation; in SLEUTH's case none of the spatial environment datasets is altered, thus the *output* tags can be left blank. The Animat stereotype also includes the *input* and *output* tags, with the same functions as those in the Spatial Environment stereotype. In the GIS context, the ending configuration of an Animat class will always be relevant.

SLEUTH has been called a "self-modifying" model for it adjusts behaviour rules during simulation. This is done to mimic logistic growth of cities, by which the rate of growth follows an S-shaped curve. To this end an additional input, Capital, is added to the model, setting this urban growth rate. Using the Global stereotype, a time series can be fed to the model; an *input* tag can again be used to parametrize access to a text file, producing at least as many successive values as those defined in the *steps* tag of the Simulation class.

Figure 5 presents the main data inputs to SLEUTH with the DSL3S constructs in the Integrated view.

SLEUTH produces urban growth by three different mechanisms, which are captured as three different behaviours in the DSL3S model; they are: (i) Dif-

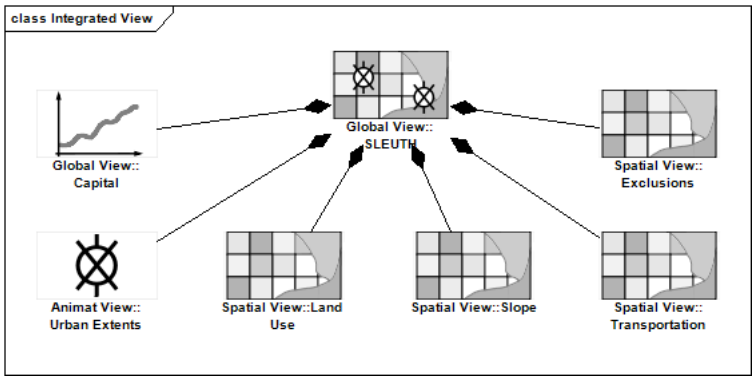

Fig. 5. Global, Spatial and State View for the SLEUTH model [16].

fusive growth - the emergence of new city centres in suitable places, a good example of an Originate type of behaviour; (ii) Organic growth - the sprawl of cities' extents around its periphery, modelled with a Replicate behaviour; and (iii) Transport lead growth - the propensity of urban areas to expand around communication pathways, also a Replicate type of behaviour. Each of these behaviours is presented in its own view in Figure 6.

The behaviour stereotypes that provide the relationships from animat to variables can be parametrized using three main tags: *minTreshold*, *maxTreshold* and *probability*. The first two set minimum or maximum values beyond which the modelled behaviour can occur, e.g. in the case of the *Diffuse.Slope* class *maxTreshold* could be set at a value of 5, signifying that diffuse growth can only occur in places where slope is below 5%. This implies that datasets like that provided as input for the *Land Use* class must be ordered in way to have all suitable land use classes coded with values higher or lower than non suitable classes. The *probability* tag defines a variable as a random deterrent of a behaviour. At each time step a random number is generated within the variable's interval of admissible values and compared with the value at the present time step and/or location, thus determining if the action should occur or not. This is the case with all the Capital behaviours, this way reducing the chances of new urban areas developing with time.

4 Summary and Future Work

DSL3S is an ongoing project to develop a Domain Specific Language for Spatial Simulations, whereby graphical models can be directly translated into ready-to-run applications. A prototype UML profile has been presented in this article that attempts to capture core spatial dynamics concepts free of application field specificities. The key concepts introduced in this profile seem to be sufficient to describe popular simulation models such as Urban Growth.

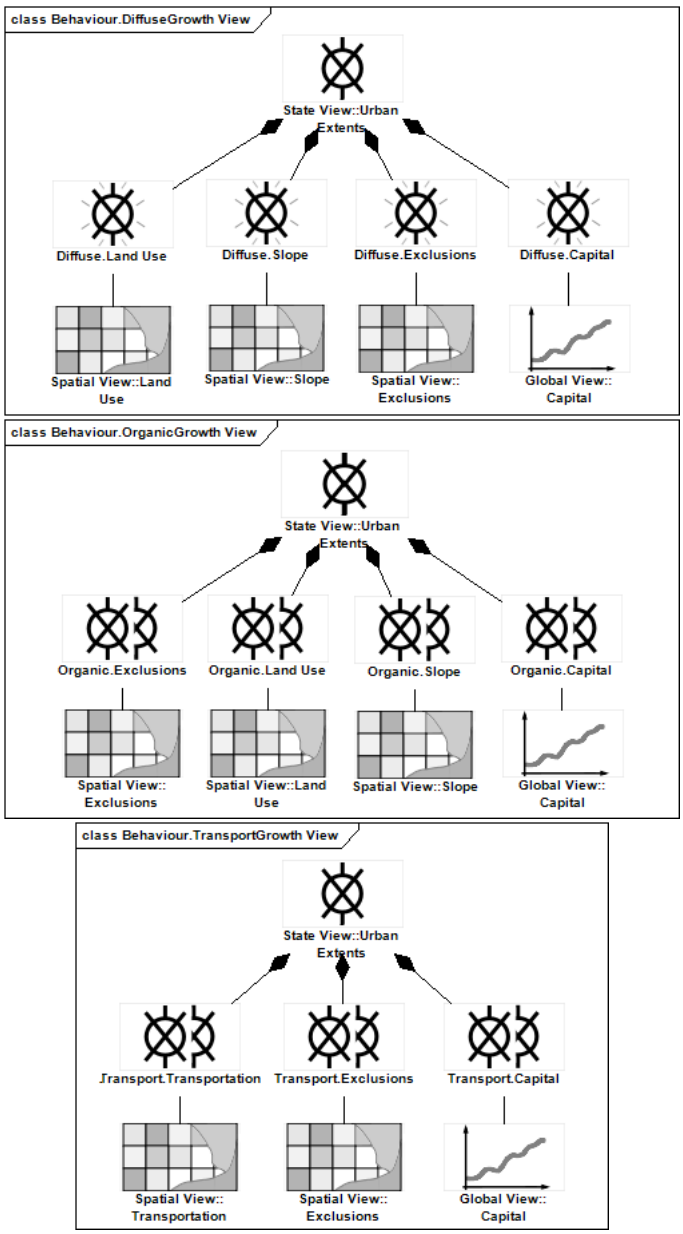

Fig. 6. Realization of the SLEUTH model [16] . Top: Behaviour.Diffuse View; center: Behaviour.Organic View; bottom: Behaviour.Transport View.

The project will now proceed with the development of code generation templates to translate DSL3S models into coded simulations, with both RePAST and MASON being considered as target code libraries. Once the coding mechanism is finished the project shall pass into an application phase, where real-life problems shall be modelled with the DSL3S.

References

1. Batty, M.: Cities and Complexity. MIT Press (2007)
2. Collier, N., Howe, T., North, M.: Onward and upward: the transition to repast 2.0. In: First Annual North American Association for Computational Social and Organizational Science Conference. Pittsburgh, Pa (2003)
3. Epstein, J.M., Axtell, R.: Growing Artificial Societies. MIT Press (1996)
4. Ermentrout, G.B., Edelstein-Keshet, L.: Cellular automata approaches to biological modeling. Journal of Theoretical Biology 160, 97–133 (1993)
5. Fall, A., Fall, J.: A domain-specific language for models of landscape dynamics. Ecological Modelling 141, 1–18 (2001)
6. Ginot, V., Le Page, C., Souissi, S.: A multi-agents architecture to enhance end-user individual based modelling. Ecological Modelling 157, 23–41 (2002)
7. Law, A.M.: Simulation Modeling & Analysis, Fourth Ed. McGraw-Hill (2007)
8. Li, X., Magill, W.: Modeling fire spread under environmental influence using a cellular automaton approach. Complexity International 8, 1–14 (2001)
9. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: A multi-agent simulation environment. Simulation: Transactions of the society for Modeling and Simulation International 82(7), 517–527 (2005)
10. Merzenich, J., Frid, L.: Projecting landscape conditions in southern utah using vddt. In: Bevers, M., Barrett, T.M. (eds.) Systems Analysis in Forest Resources: Proceedings of the 2003 Symposium. pp. 157–163. U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station, Portland, OR (2005)
11. Messina, J.P., Walsh, S.J.: The application of a cellular automaton model for predicting deforestation. In: Proceedings of the 4th International Conference on Integrating GIS and Environmental Modelling. Banff, Canada (2000)
12. Minar, N., Burkhart, R., Langton, C., Askenazi, M.: The Swarm simulation system: a toolkit for building multi-agent simulations (1996)
13. Mladenoff, D.: Landis and forest landscape models. Ecological Modelling 180(1), 7 – 19 (2004), modelling disturbance and succession in forest landscapes using LANDIS
14. OMG: Uml 2.0 specification. http://www.omg.org/spec/UML/2.0/ (July 2005), retrieved January 2011
15. Railsback, S.F., Steven, L.L., Jackson, J.K.: Agent-based simulation platforms: Review and development recommendations. Simulation 82 Issue 9 (2006)
16. Silva, E.A., Clarke, K.C.: Calibration of the sleuth urban growth model for lisbon and porto, portugal. Computers, Environment and Urban Systems 26, 525–552 (2002)
17. Weiss, G. (ed.): Multiagents Systems: a Modern Approach to Distributed Artificial Intelligence. MIT Press (1999)
18. Wilson, S.W.: The animat path to ai. In: Meyer, J.A., Wilson, S. (eds.) From Animals to Animats. pp. 15–21. MIT Press, Cambridge, MA (1991)
19. Wuensche, A., Lesser, M.: The Global Dynamics of Cellular Automata. Addison-Wesley (1992)
20. Yi, W., He, B.: Applying sleuth for simulating urban expansion of beijing. In: 2009 International Joint Conference on Artificial Intelligenc. vol. 2, pp. 652–656 (2009)