An Operational Meta-Model for Handling Multiple Scales in Agent-Based Simulations

Duc An Vo, Alexis Drogoul, Jean-Daniel Zucker

An operational meta-model for handling multiple scales in agent-based simulations

Duc-An Vo[1, 2], Alexis Drogoul[1, 2] and Jean-Daniel Zucker[1, 2]

[1] IRD, UMI UMMISCO 209, 32 avenue Henri Varagnat, 93143 Bondy, France
[2] IFI, MSI, UMI 209, ngo 42 Ta Quang Buu, Ha Noi, Viet Nam
voducanvn@yahoo.com, alexis.drogoul@gmail.com, jdzucker@gmail.com

Abstract-**There is a growing interest for multi-scale agent-based modeling. The research community has made several efforts in proposing agent-based simulators or meta-models, which accommodate for multi-scale agent-based modeling. Unfortunately, the proposed simulators are often tight to a specific model. And the proposed meta-models exist only as formal proposals without a concrete implementation supported by an operational modeling language. The lack of a common operational meta-model and an associated operational modeling language raises the question of reusability and increase the danger of "re-inventing the wheel" when one develops new models. Moreover, the operational semantics of such multi-scale models may differ without having defined a solid theoretical ground. Our research aims at proposing a generic meta-model for multi-scale agent-based modeling. In our approach, we revisit the fundamental notions of agent modeling (agent/environment/scheduler) in order to support a recursive representation of these notions in an agent-based model. To test the operational semantic of the proposed model, we implement this meta-model as an agent-based modeling language in the GAMA simulation platform. This paper presents such a meta-model and how its concepts can be used to formalize multi-scale agent-based models.**

Keywords-**Agent-Based Modelling; Simulation; Multi-Scale; GAMA**

I. INTRODUCTION

There is a growing need in the development of multi spatial and temporal scale agent-based models (ABM). Developing these models requires the possibility to simultaneously represent several spatial and temporal scales in an ABM. The spatial scale is represented as environment in which agents live. The temporal scale is represented as the execution frequency of agents, which is controlled by a scheduler. So the modeler needs in fact the possibility to represent at the same time multiple environments and schedulers when developing a multi-scale ABM.

The research community has made several efforts in proposing agent-based simulators or meta-models, which accommodate for multi-scale agent-based modeling. Unfortunately, the proposed simulators are often tight to a specific model. And the proposed meta-models exist only as formal proposals without a concrete implementation supported by an operational modeling language. The lack of a common operational meta-model and an associated operational

modeling language raises the question of reusability and increase the danger of "re-inventing the wheel" when one develops new models. Moreover, the operational semantics of such multi-scales models may differ without having defined a solid theoretical ground.

To address this issue, our work aims at proposing a meta-model for a multi-scale agent-based modeling. This meta-model is implemented as a modeling language in the GAMA platform [Drogoul 2011]. After a quick review of current proposals for multi-scale ABM development, we will show how current proposals suffer from multi-scale representation through a simple imaginary multi-scale ABM. After that we will introduce our meta-model. Then we will show how the concepts proposed in the meta-model can be employed to represent an example multi-scale ABM. Finally, we will conclude the article and give some perspective for this work.

II. STATE OF THE ART OF MUTLI-SCALE AGENT-BASED MODELING

Modeling languages of current agent-based simulators such as NetLogo [Wilensky 1999], Repast [North 2007], … don't explicitly support the multi-scale representation in an ABM. These languages "assume" that all agents in a model share only one environment and scheduler. Hence this is an obstacle for the modeler when he needs develop a multi-scale ABM which requires to explicitly take into account agents of different spatial and temporal granularity.

While existing simulators' modeling language fail to answer the question of multi-scale representation, some modelers develop dedicated simulators for their multi-scale ABMs such as SimulBogota [Gil-Quijano 2007], RIVAGE [Servat 1998], …. that are tight to the specific application domain, which raises the problem of the reusability of the simulator.

Several researchers try to propose meta-models, which claim to support multi-scale agent-based modeling. But these meta-models exist only as formal specification without concrete implementations as ready-to-use modeling languages [Langlois 2010], [Gignoux 2011]. Some meta-models propose good theoretical multi-scale concepts but to our investigation it isn't feasible to operationalize all these concepts under a modelling language [Langlois 2010], [Gignoux 2011]. Others don't support spatially explicit representation and don't take

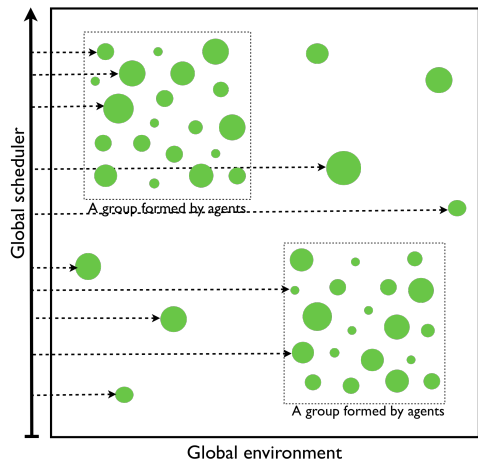into account the geometric shape of agents such as [Picault 2010].

Figure 1 - Imaginary multi-scale ABM with mono-scale representation.

To make the problem clear, let consider an imaginary ABM depicted in figure 1. The outer square represents the bound of the global environment in which agents live. Green circles are agents. The arrow in the left-side of the outer square stands for the scheduler of the simulation. Dotted arrows from the global scheduler to each agent represent the fact that the execution of these agents is controlled by this scheduler. We draw only some representative dotted arrows in order not to complicate the figure. The two inner squares may either represent emergent structures formed by groups of green circle agents or another kind of agent belonging to a different spatial & temporal scale compare to the green circle agents. So this model contains simultaneously agents belong to different scales.

Figure 1 shows the fact that green circle agents, belonging or not to the group indicated by the inner square, all share the same global environment and scheduler. In fact, an agent stands alone may exhibit different dynamic compare to that same agent when it belongs to a group [Vo 2010]. Current modeling languages don't explicitly support the representation of this dynamic change. Moreover, these languages support neither the representation of emergent structures in an ABM nor the simultaneous representation of agents belonging to different scale.

III.   META-MODEL

The methodology employed in this work is to 1) revisit the fundamental concepts of agent-based modeling (agent/environment/scheduler) and 2) work on several multi-scale agent-based modeling projects to draw common multi-scale concepts between these projects. The review of the fundamental concepts gives us a starting point and helps to find out what are missed in order to be able to represent a multi-scale ABM. Working on several multi-scale ABMs helps us to find out what concepts are needed to represent these models. These multi-scale ABMs are also useful testcases to verify the generality of our meta-model and modeling language.

From the description of the failure of current approaches in representing multi-scale ABM depicted in figure 1, we propose a meta-model which is capable of representing the imaginary multi-scale ABM.
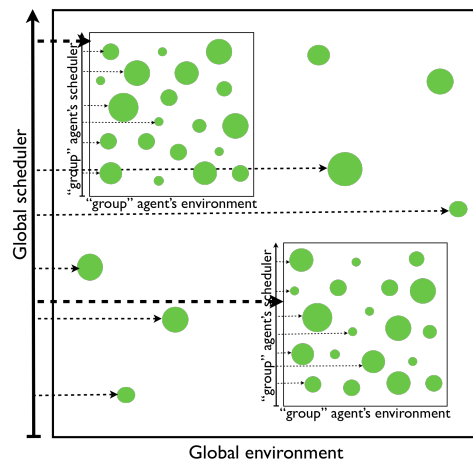

Figure 2 - Imaginary multi-scale ABM with multi-scale representation.

Shortly, the meta-model supports the recursive representation of spatial and temporal scale. Agents are organized as hierarchy. An agent has a scheduler to schedule its components (micro-agents) and an environment in which its micro-agents can live. From the figure 2, we see that two inner squares are explicitly represented as agents in the model. They have their own scheduler to control the execution of their micro-agents and an environment in which their components can live.

A.   Description

The UML class diagram in figure 3 shows the proposed meta-model. Classes represent concepts. Links between classes represent relationships between concepts. This section describes these concepts and the relationships between them.

- **Species** defines attributes and behaviours of a class of same type agents. The relation "*specializes*" from species to itself shows that a species can be a sub-species of another species. Sub-species inherits all the attributes and behaviours of its parent. Sub-species can re-define attributes and behaviours of its parent. The relation "*contains*" from species to itself indicates that a species can contains other species. The outer-species is considered as macro-species compared to its set of inner-species. The inner-species is considered as micro-species compared to its outer-species.

- **Agent** represents a simulated entity of the complex system in the simulation. It is initialized from the information specified in its species. Agents of the same species share common attributes and behaviours. An **Agent** implements several interfaces (*IAgent*, *IScheduler*, *IHost*, *IGeometry*) so it has multiple roles.

- **IAgent** encapsulates all attributes and behaviours of an agent.

- **IGeometry** indicates that an agent has a geometry representing its shape. It helps to represent the spatially explicit agent.

- **IHost** permits an agent to play the role of a macro-agent. A macro-agent can contains others agent as its micro-agents.

- **IScheduler** lets an agent play the role of a scheduler. A scheduler represents a temporal scale and is used to step the agents in the simulation.

- **Population** is a collection of same species agents. The "*houses*" relation indicates that a population is hosted in an **IHost**, which is a macro-agent. Agents in populations are micro-agents which has access to its macro-agent thanks to the relation with **IHost**. The "*schedules*" relation with **IScheduler** reflects the fact that agents in a population are scheduled by the corresponding macro-agent's scheduler. This relation gives the possibility for an agent to define a temporal scale to control the execution frequency of its micro-agents.

- **IControl** defines the control type of agent. The control can be considered as a behaviour model which decides how an agent executes his behaviour in the simulation. Popular behavior models are reflex based, finite state machine, task based, … . The "*specified by*" relation to **Species** says that the type of control is specified in the species definition. The "*provides*" relation from Population models the fact that all agents in a population share a control. This control is used to step all the agents of this population according to the "executes" relation to **IAgent**.

- **ITopology** represents the spatial organization of the environment in which agents live. It conditions how agents of a population move. There are different types of topology such as grid topology, graph topology, … which are used to model different types of spatial organization. The "*bounded by*" relation shows that a topology is constructed and bounded from the geometric shape of an agent. The "*uses*" relation with **Population** indicates that agents in a population move respecting an associated topology. So we can say that agents of a population move in the geometric shape of its macro-agent. The shape of an agent can play the role of an environment in which other agents can live. This offers the possibility to represent simultaneously agents belonging to different spatial scales.
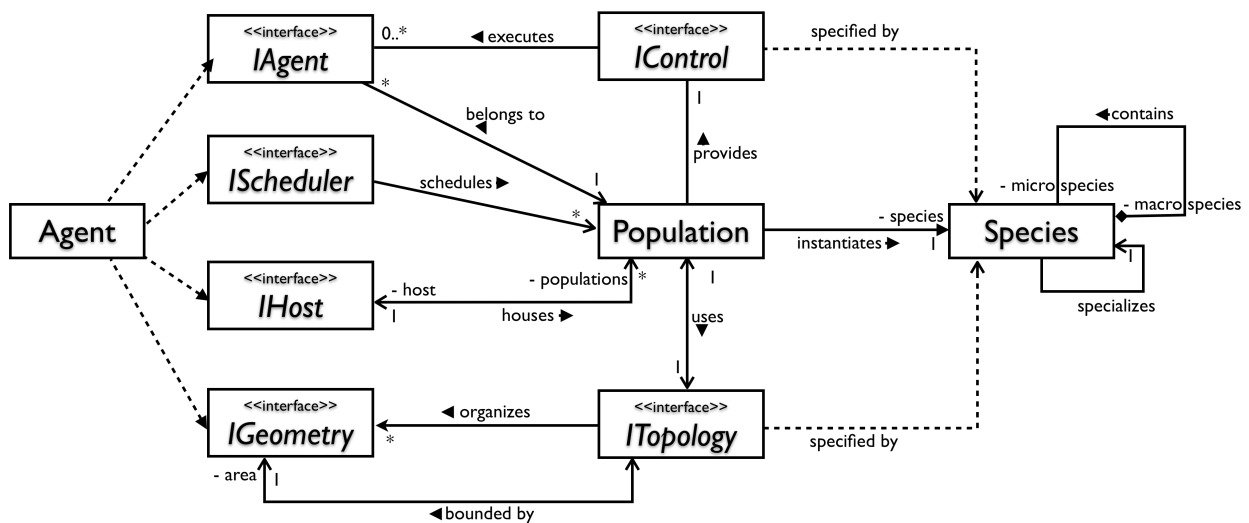

Figure 3 - Meta-model for multi-scale agent-based modeling.

B. Multi-scale model formalization

According to the previous description, we see that an agent can offer an environment, through its shape, in which other agents can live and a scheduler to control the execution of its micro-agents. This possibility of recursive representation of environment and scheduler enables the modeler to develop multi-scale agent-based models. We propose to add a special agent, which plays the role of a root agent. This agent can be named as *root agent*, *global agent*, *world agent* or *system agent*, etc. It offers a global environment in which its micro-

agents can live. It offers also a scheduler with the finest frequency used to step its micro-agents. Beginning with the root agent then progressively refining the spatial and temporal scale, the modeler can obtain a multi-scale model representation.

Figure 4 shows an example model with two scales. The "root" species defines attributes and behaviors of the root agent. This species represents a global agent playing the role of a context agent which is supplied automatically to every simulation. At level 1, we have two species ("species 1" and

"species 2") as micro-species of the "root" species. At level 2, we have 1 species ("species 3") which is a micro-species of "species 1" species.
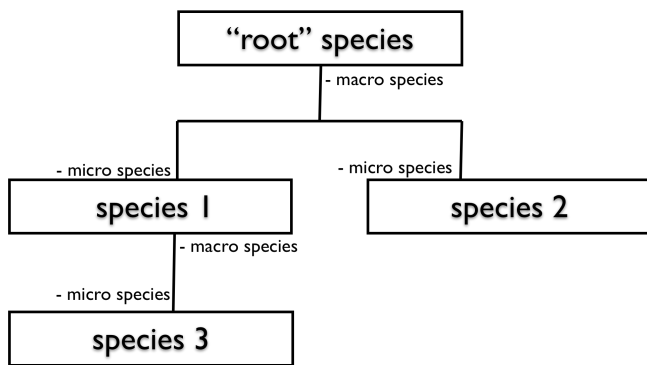

Figure 4 – Multi-scale model example.

Any agent can become a macro-agent of other agent. And vice-versa, any agent can join the list of micro-agents of other agent. So the modeler needs the possibility to model the fact that 1) an agent captures other agents as its micro-agents 2) an agents doesn't consider some agents as its micro-agents anymore thus releases them.

1) Capture agents

An agent can make agents of other species become its micro-agents by capturing them. We define the "capture" operation permitting the modeler to do this. When capturing an agent, the modeler can specify the following information:

a) The new species that the captured agent will become. This new species is a micro-species defined in the species of the agent doing the capture. It is also a sub-species of the species of the agent being captured. So when an agent is captured to become a micro-agent, it will change species. Figure 5 illustrates clearly this idea. We add the new "species 4" species to the figure 4's example model. "species 4" species is a micro-species of "species 1" species and a sub-species of "species 2" species. So in the context of a "species 1" agent, the modeler can capture a "species 2" agent and make this agent become a micro-agent of the "species 1" agent with "species 4" as its new species. As "species 4" species is a sub-species of "species 2" species, so in the definition of the "species 4" species, the modeler can 1) change "species 2" species' attributes and behaviors; 2) add new attributes and behaviours. This helps to model the fact that an agent can change its behavior when becoming micro-agent of other agent [Vo 2010].

b) The agent is spatially captured or not. When capturing an agent, the modeler can decide whether it is spatially captured or not. An agent is considered as spatially captured if has a coordinate in the environment bounded by the shape of its macro-agent. Being spatially captured means that the agent is spatially managed by its direct macro-agent and can not exit the macro-agent's environment unless it is explicitlt released by the macro-agent. An agent can also be captured but not-spatially. In this case, it is also considered as a micro-agent of the agent directly captured it but it is spatially

managed by the agent that spatially manages its macro-agent. The "root" agent spatially manages itself.

Figure 5 – A "species 2" agent can change to a "species 2" agent when being captured by a "species 1" agent.

c) The agent is temporally captured or not. The modeller can also decide whether to capture an agent temporally or not. When an agent is temporally captured, it is temporally managed by it macro-agent and its execution will be controled directly by the scheduler of its macro-agent. If not, its execution is controled by the scheduler of the agent that manages that execution of its macro-agent. The "root" agent temporally manages itself.

2) Release agents

A macro-agent can decide to release certain micro-agents appropriately. We define the "release" operation allowing the modeler to represent this fact. When being released from a macro-agent, the corresponding agents will 1) take the macro-agent of its current macro-agent as its new macro-agent; 2) change the species to the species right before being captured by the current macro-agent. Figure 5 can illustrate this idea. For example, an agent of "species 4" species is a micro-agent of a "species 1" agent. With the "release" operation, the modeler can release this agent from the "species 1" agent. In this case, the released agent will take the root agent as it macro-agent, change its species to "species 2" species. If it is spatially and/or temporally captured by the "macro 1" agent, it will no longer be. It will be spatially and temporally captured by its future macro-agent (root agent in this case).

Reader may have question for the "species 3" species in figure 3. Can a "species 1" agent release an agent of "species 1" species? We see that the "species 3" species is not a sub-species of any other species. So the "release" operation can't determine which future species to change to when a "species 1" agent releases a "species 3" agent. To make the "release" operation work for "species 3" agent, we just have make the "species 3" species be a sub-species of another species at the same level of "species 1" species. It means that parent of "species 3" species is a direct sub-species of "root" species as "species 1" species. But in this case, the modeler may have no need to release the "species 3" agents from "species 1" agents, so he doesn't define a parent for "species 1" species.

Together with the recursive representation of spatial and temporal scale offered by the meta-model, "capture" and "release" operations bring the possibility to model the dynamically changing of scale and behavior of agent during the simulation. This provides the modeler with the necessary framework to design multi-scale agent-based model.

C. Implementation of a modeling language

To test the applicability of the meta-model in an operational context, we implement this meta-model as the GAML modeling language of the GAMA simulation platform [Taillandier 2010], [Drogoul 2011]. Thanks to this meta-model, the GAML language becomes capable of helping modeler to develop multi-scale agent based models. We are testing this language in several multi-scale agent-based modeling projects. Some of the current projects are: modeling the evacuation activity in Nha Trang city responding to a tsunami alarm [Nguyen 2011]; modeling the propagation of brown plant hopper in the Me Kong Delta region in the south of VietNam [Vo 2011]; modeling the propagation of disease on Catfish [Ho 2010], modeling the development of trees and bosquets in a savanna [Gignoux 2008-2011]. In the context this article, we don't go to the detail of the GAML language. Instead, we will show how the concepts of this meta-model can be used to model the evacuation activity in Nha Trang city.

IV. FORMALIZING A MULTI-SCALE AGENT-BASED MODEL

A. Problem description

Nha Trang is a coastal city in the central of Viet Nam. It is susceptible, one day, to be touched by a tsunami resulting from an earthquake original from the Philippines. In order to decrease the potential victims, the Vietnamese government would like to prepare in advance the evacuation strategies, which are the most efficiency possible.

Figure 6 – Proposal of evacuation panel to be placed in Nha Trang city (source: VAST).

An agent-based model (figure 7) using the GAML modeling language is being built in order to assist the evaluation of different evacuation strategies. We would like to set up a system of evacuation panels in Nha Trang city (figure 6) and build a system of shelters in the city. When the tsunami alerting alarm is raised, citizens will follow the direction indicated by evacuation panels in order to reach the safe shelters. Results of different simulation outcomes are among the considered factors helping to set up the evacuation panel configuration and build the shelter system.

B. Model and simulation

This model includes the following main species:

- **citizen** represents citizens of the city.
- **road** represents roads in the map. The map of Nha Trang city is loaded to the simulation from the GIS (Geographical Information System) data.
- **shelter** represents safe buildings in which citizen can shelter.
- **evacuation panel** represent evacuation panels placed at different crossroads.

The road network of Nha Trang city is loaded from the GIS data. A number of shelters are placed in the city. Some crossroads are equipped with evacuation panels indicating direction to these shelters. The simulation period is the time from the tsunami alarm is raised to the moment when the tsunami touches the city. The purpose of the simulation is to discover under what "configuration" (of shelters, of evacuation panels placement), the number of citizens reach safe place will be maximum in the pre-specified simulation period.

Experiments of different simulation scenarios have been performed on this model to quantify the number of citizens arriving at the shelters before the tsunami attacks the city [Nguyen 2011]. However, once these experiments include the entire population of Nha Trang (estimated at 350,000 according to the last census [Wikipedia 2011]), performance issue becomes a clear obstacle. With a huge number of agents, the speed of simulation becomes unacceptable slow.

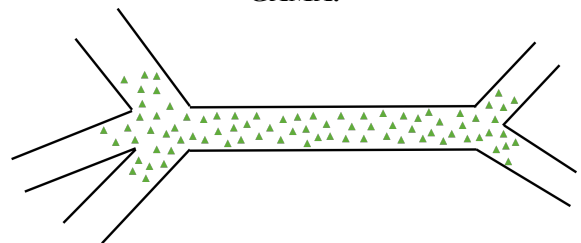**Figure 7 – Evacuation simulation of Nha Trang city in GAMA.**

Figure 8 – A road with moving citizens.

In order to speed up this simulation without influencing the simulation outcome, we propose to divide a road into three patches (figure 9). Patch 1 and 3 are two ends of the road, near the crossroads. Within these two patches, each citizen is simulated individually as agent. When reaching crossroad, they observe evacuation panels in order to follow the right direction towards the shelters. Patch 2, in the middle, occupies the most part of the road. Movement of citizens in patch 2 is simulated using a macroscopic model of pedestrian flow [Hughes 2002].

We need to take into account the interactions between patch 1 & patch 2, patch 2 & patch 3. When a citizen in patch 1 reaches the border with patch 2, the "capture" operation helps to model that fact that a "road" agent captures a citizen agent as

its micro-agent. Species of the captured agent will change from "citizen" to "road_citizen" (figure 10). The citizen is spatially and temporally captured by the road agent. It means that: 1) the captured citizen has a coordinate in the "patch 2" zone which is not important anymore 2) its execution is managed by the scheduler of the "road" agent. The "road_citizen" species overrides all the behaviors of "citizen" species, makes them empty. As all the movement of the captured citizens in patch 2 has already been taken into account by the road agent. According to [Hughes 2002], the movement of pedestrians within the road can be simulated using the macroscopic model without decreasing the output result compare to microscopic approach such as ABM. So the fact that we don't simulate individually all the pedestrians within patch 2 can help to save the computing resource but at the same time doesn't influence the simulation outcome. When a "patch 2" agent "sees" that it needs to release certain micro-agent citizens, it will do this thanks to the "release" operation. A "road_citizen" agent will change to a "citizen" agent. The newly released agent is put at the beginning of patch 3 and continues to move towards the crossroad where it can see the evacuation panel.
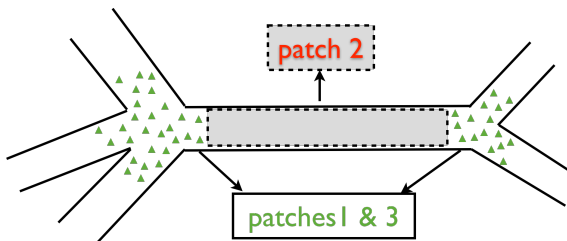

Figure 9 – Three patches of a road.

This model now becomes a multi-scale ABM with the following species' conceptual structure.
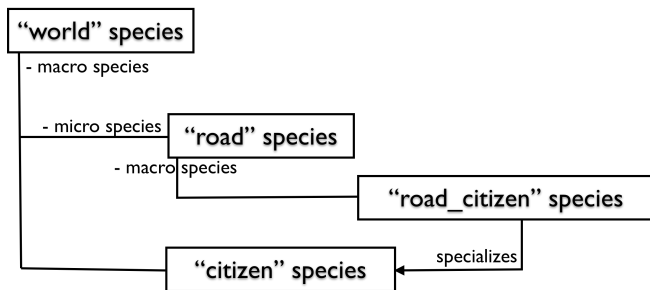

Figure 10 – Multi-scale species structure of the evacuation model.

V.   CONCLUSION

The proposed meta-model permits a recursive representation of species, agent, environment and scheduler in an agent-based model. The two operations "capture" and "release" help to model the dynamic change of scale and behavior of agents during the simulation. Thus the meta-model and these two operations supply the modeler with the necessary utility to develop multi-scale ABM.

An agent-based modeling language is being implemented in the GAMA platform [2] in order to test the applicability of the meta-model in the operational context. We are also testing this language in several multi-scale ABMs such as [Nguyen 2011], [Vo 2011], [Ho 2010]. Testing the modeling language in several ABM projects helps to improve its generality and provides a direct feedback to refine the meta-model.

VI.   ACKNOWLEDGEMENT

REFERENCES

[1] A. Drogoul. GAMA platform: http://gama-platform.googlecode.com/. UMI 209, UMMISCO, IRD, 32 av. Henri Varagnat, 93143 Bondy cedex, France, 2011.

[2] J. Gignoux, I.-D. Davies, S.-R. Flint and J.-D. Zucker, "The Ecosystem in Practice: Interest and Problems of an Old Definication for Constructing Ecological Models", Ecosystems, Spriner Science+Business Media, LLC, 2011.

[3] J. Gignoux, 3Worlds project: http://threeworlds.biologie.ens.fr/, 2008-2011.

[4] J. Gil-Quijano, M. Piron and A. Drogoul, "Mechanisms of automated formation and evolution of social-groups : A multi-agent system to model the intra-urban mobilities of Bogota city," chapter 12, pages 151-169. Idea Group Inc., 2007.

[5] P. Langlois, E. Daudé, B. Blanpain, E. Sapin. "AOC, une ontologie formelle pour la modélisation de système complexes en géographie". Conférence internationale de Géomatique et Analyse Spatiale, Toulouse, France, 2010.

[6] V.-T. Ho, "Simulating the propagation of Catfish Diseases in Pon system.", Master thesis, Can Tho University, 2010.

[7] R.-L. Hughes, "A continuum theory for the flow pedestrians.", Transportation research Part B: Methodological 36 (6), 507-535, 2002.

[8] T.-N.-A. Nguyen and al. "Hybrid Equation-based and Agent-based Modeling of Crowd Evacuation on Road Network". DOCMAS workshop, Taiwan. 2011.

[9] M.-J. North, E. Tatara, N.T. Collier, J. Ozik, "Visual agent-based model development with Repast Simphony. In: Proceedings of the agent 2007 conference on complex interaction and social emergence". Argonne: Argonne National Laboratory, 2007.

[10] S. Picault, P. Mathieu, Y. Kubera, "PADAWAN, un modèle multi-échelles pour la simulation orientée interactions", JFSMA 2010, Valence, 2010.

[11] D. Servat, E. Perrier, J.-P. Treuil, and A. Drogoul, "When agents emerge from agents: Introducing multi-scale view-points in multi-agent simulations." In J.S Sichman, R. Conte, and N. Gilbert, edtors, MASB, volume 1534 of LNCS, pages 183)198. Springer, 1998.

[12] P. Taillandier, D.-A. Vo, E. Amouroux and A. Drogoul, "GAMA: a simulation platform that intergrates geographical information data, agent-based modeling and multi-scale control". PRIMA 2010, Kolkata, India, 2010.

[13] D.-A. Vo, A. Drogoul, J.-D. Zucker and T.-V. Ho, "A Modelling Language to Represent and Specify Emerging Structures in Agent-Based Model", PRIMA 2010, KolKata, India, 2010.

[14] T.-T. Vo and al, "Modeling brown plant hopper population dynamics under effect of certain environmental factors in Mekong Delta region, VietNam.", IEEE ICIA Technology, 2011.

[15] Wikipedia, "Nha Trang city, http://en.wikipedia.org/wiki/Nha_Trang/" [access September-2011]

[16] U. Wilensky, NetLogo: http://ccl.northwestern.edu/netlogo/. CCL, Northwestern University, Evanton, 1999.