MULTI-AGENT MULTI-LEVEL MODELING – A METHODOLOGY TO SIMULATE COMPLEX SYSTEMS

Jean-Baptiste Soyez[(a)], Gildas Morvan[(b)], Rochdi Merzouki[(c)],Daniel Dupont[(d)]

[(a,b,c,d)]Univ. Lille Nord de France, 1bis rue Georges Lefèvre 59044 Lille cedex, France
[(a,c)]LAGIS UMR CNRS 8146 École Polytechnique de Lille Avenue Langevin 59655 Villeneuve d'Ascq, France
[(b)]LGI2A, Univ. Artois Technoparc Futura 62400 Béthune, France
[(a,d)]HEI Hautes Études d'Ingénieurs 13 rue de Toul 59046 Lille cedex, France

[(a)]Jean-Baptiste.Soyez@polytech-lille.fr, [(b)]gildas.morvan@univ-artois.fr, [(c)]rochdi.merzouki@polytech-lille.fr, [(d)]daniel.DUPONT@hei.fr

ABSTRACT
This article deals with the conception, modeling and simulation of complex systems, represented at different *levels* of analysis with respect to the agent-based modeling (ABM) paradigm and more precisely on the generic meta-model IRM4MLS. A methodology, using IRM4MLS, is proposed to save computational resources in multi-level agent-based simulations, representing only the relevant elements. It means that the structure of agents can be modified during simulation, by temporarily aggregating, removing or approximating their characteristics to maximize their life-cycles.

Keywords: multi-agent modeling, multi-level modeling, simulation, influence/reaction model

1. INTRODUCTION

This article deals with the conception, modeling and simulation of complex systems, represented at different levels of analysis. A level can be defined as a point of view on the studied system and its relations to other points of view. Therefore, in most real world applications, a level encapsulates the processes executed at a given spatio-temporal scale; the term multi-scale is also used. This work relies on the agent-based modeling (ABM) paradigm and more precisely on the generic meta-model IRM4MLS introduced by Morvan et al. (2010), based on the influence/reaction principle (Ferber & Muller 1996).

According to Guijano and al. 2009, three important issues in *classic* (or flat) ABM can be solved by multi-level agent-based models (MAM). (1) Some complex systems cannot be understood without integrating knowledge that is ontologically distributed over multiple levels of organization. In other words, system behavior cannot, using the available knowledge, be described with a purely emergent (or bottom-up) approach. Examples of such models can be found in Morvan and al. 2008 and Morvan and al. 2009. (2) Many distributed systems are characterized by a macroscopic behavior that becomes remarkable when the number of entities is important. Indeed, *"agentifying"* this behavior can be useful (Servat and al. 1998). (3) Simulated entities can be reflexive, i.e., able to reason on social facts (similarities with other agents, group involvement, etc.) such as in Gil Quijano and al. 2009 and Pumain and al. 2009.

In this article, a methodology using IRM4MLS, is proposed to tackle complexity issues of MAM which have to simulate many entities with complex interactions. An important aim of this methodology, among others, is to save computational resources, representing only the relevant elements of a simulation when they are needed, i.e., lighten the representation of agents and their life-cycles. The main idea is to identify agent characteristics (state variables, available actions and decision processes, etc.) that can be modified at run-time. It means that the structure of agents can be modified during simulation, by temporarily aggregating, removing or approximating their characteristics to maximize their life-cycles.

2. RELATED WORKS

The works on multi-level agent-based modeling are related to other approaches that view simulations of complex systems as *societies of simulations*.

The High Level Architecture (HLA) is a general purpose architecture for distributed simulations computer simulation systems. Using HLA, computer simulations can interact (that is, to communicate data, and to synchronize actions) to other computer simulations regardless of the computing platforms. The interaction between simulations is managed by a Run-Time Infrastructure (RTI).

Holonic multi-agent systems (HMAS) can be viewed as a specific case of multi-levels multi-agent-systems (MAS), the most obvious aspect being the hierarchical organization of levels. However, from a methodological perspective, differences remain. Most of holonic meta-models focus on organizational and methodological aspects while MAM is process-oriented. HMAS meta-models have been proposed in various domains, e.g., ASPECTS (Gaud and al. 2008) or PROSA (Van-Brussel et al. 1998.). Even if MAM and HMAS structures are close, the latter is too constrained for the target application of this work.

Multi-Resolution Modeling (MRM) (Davis andt al. 1993) which is the joint execution of different models of the same phenomenon within the same simulation or across several heterogeneous systems, can inspire our approach if the different models are at different levels. The consistency symbolizes the amount of essential information lost during the passing between models and it is a good tool to test the quality of this approach.

Navarro and al. (2011) present a framework to dynamically change the level of detail in an agent-based simulation. That is to say, represent in detail only which is needed during simulation, to save CPU resources and keep the consistency of the simulation. But this framework is limited because all levels form a meshed hierarchy, without the possibly of having two different levels at the same scale and communication between levels is not explicitly defined.

In the section 3, the main concepts used in this article (ABM and IRM4MLS) are introduced. This presentation emphasizes our vision of agent architecture in the context of IRM4MLS. In the section 4, 3 methods to save computational resources (RAM and CPU) in models based on IRM4MLS are introduced. In the conclusion, we sum-up the main interests of our method and its perspectives.

3. MAIN CONCEPTS

3.1. Agent-based modelling and simulation

The multi-agent formalism is straightforward. Each entity of the studied system possesses an equivalent entity (or *computational agent*) in the computer representation. It is then easier to apprehend than mathematical models.

Jacques Ferber (1995), p. 12 gives a definition of what is an agent: "*We call agent a physic or virtual entity which*

a) can act in an environment,

b) can directly communicate with others agents,

c) is pushed by a set of tendencies(represented by some goals or a satisfaction function or a survival one),

d) possesses its own resources,

e) is able to perceive (but in a limited way) its environment,

f) has a partial representation of its environment (eventually none),

g) has some capacities and proposes some service,

h) can eventually reproduces itself,

i) whose behavior tend to satisfy its objectives, considering his resources and available capacities, also considering its perception, its representations and the received communications." (translation from french by authors)

An agent can be seen as an entity composed of a non-observable internal state (e.g., its beliefs, desires and intentions in the BDI architecture) and an observable external state (e.g., its position, velocity, acceleration and direction in situated multi-agent systems). In multi-agent based simulations, the execution of agents is scheduled (cf. Fig. 1).

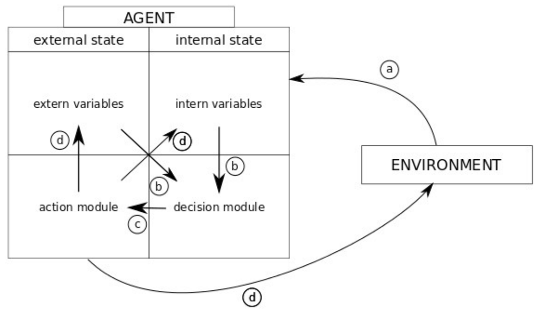

Figure 1: Basic Structure of an Agent and his life-cycle

The life-cycle of an agent can be described as follows:

a) the agent senses its environment to construct percepts,

b) the decision module selects the action to perform

c) from its internal and external states

d) the agent acts (in the influence reaction model, produces influences in its environment.

Drogoul and al. (2002) highlight that in ABM, 3 different kinds of agent are used: *"real agents"*, that can be observed in the studied system, *"conceptual agents"*, i.e., formalizations real agents with respect to MAS concepts (communications, interactions, etc.) and finally "computational agents", i.e., executable implementations of conceptual agents on the target simulation platform. This precision is important: in the following a *same real agent* will be successively represented by *different computational agents*.

3.2. IRM4MLS

IRM4MLS is a MAM meta-model proposed by Morvan and al. 2011. It relies on the influence / reaction model (Ferber & Muller 1996) its extension to temporal systems, IRM4S (Michel 2007). Beside its generality, an interesting aspect of IRM4MLS is that any valid instance can be simulated by proposed algorithms (Soyez and al. 2011). Only the main aspects of IRM4MLS are presented in this section. Readers interested in a more exhaustive presentation may refer to referenced publications.

A MAM is characterized by a set of levels, L , and relations between levels. Two types of relations are considered in IRM4MLS: influence (agents in a level l are able to produce influences in a level $l' \neq l$) and perception (agents in a level l are able to perceive the state of a level $l' \neq l$). These relations are respectively formalized by two digraphs, $\langle L, E_I \rangle$ and $\langle L, E_P \rangle$ where E_I and E_P are sets of edges, i.e., ordered pairs of elements of L . The dynamic set of agents at time t is denoted $A(t)$. $\forall l \in L$, the set of agents in l at t is $A_l(t) \subseteq A(t)$. An agent acts in a level iff a subset of its external state belongs the state of this level. An agent can act in multiple levels at the same time.

Environment is also a top-class abstraction. It can be viewed as an agent with no internal state that produces "natural" influences in the level (Fig. 2).

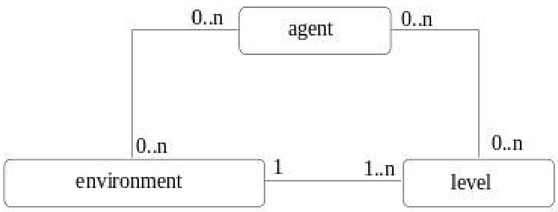

Figure 2 : Central Concepts of IRM4MLS (cardinalities are specified with the UML way)

The scheduling of each level is independent: models with different temporalities can be simulated without any bias. On an other hand, it permits to execute only the relevant processes during a time-step.

A major application of IRM4MLS is to allow microscopic agents (members) to aggregate and form-up lower granularity agents (organizations). It has to be noticed that level does not necessarily means scale: it can be useful to create multiple levels at a same scale.

4. THE METHODOLOGY

4.1. General principle

Agent-based modeling possesses many advantages, but it can be difficult to be run in a simulation because the agent execution is often greedy in computer resources, particularly in used CPU time and memory.

A simple idea is to keep active in the simulator the minimum number of variables and processes and update these elements with the lower frequency (only when it's necessary). All unused elements, at a given moment, by the simulator can be write on the hard disk and delete of the simulation. When the simulator need these elements it can read and integrate them again. This is useful only if the gain spawn by the smaller quantity of data to be managed and it smaller updating frequency is more important than the loss of time generated by the reading and writing of data on the hard disk.

Also, it's not necessary to always know with the same precision some data, that's why these data can be aggregated (Lucia 2010) or approximate. There are many methods or algorithms to apply these three mechanisms : approximation, aggregation or reading/writing data on hard disk and free random access memory. We do not propose new algorithms to dynamically determine which active data can be processed by these methods.

This work takes place in the InTrade european project, which deals with the Autonomous Intelligent Vehicles (AIV) traffic flow in the major european ports. ScannerStudio is a real time simulator created for InTrade. This makes us work on the real time problematic where the proposed methods to lighten a simulation have a strong sens. The example cited below are inspired and can be implemented in ScannerStudio.

4.2. Agents resources

Agent structure is rather an heavy thing. Modelers of a simulation, generally, conceive agents with the four parts described above, which integrate all elements which can be needed at one moment of the simulation. We propose to dynamically adapt the structure of an agent to only represent what is necessary at a given time. This includes the used agent resources or the processes whom the modelers want to observe. Description of the life cycle and the structure of an agent and it modeling composed of four parts permit us to apprehend the dependencies between these parts. A modeler can see, during the modeling or the simulation phases, that these parts can be decomposed into relatively independent sub-parts. And to take a decision the sub-parts of the decision module don't called the same variables (internal or external), the sub-parts of the decision module don't activate the same actions and each action modify a different set of variables.

In his book Calvez (1990) mentions some decomposition methods which can be applied to specify a system to model it. Applying these decomposition methods, which can be functional, structural or modal, on a agent permit to determine the sub-parts of the four parts of this agent. Functional decomposition decompose the agent mission in a number of independent functions. For example an intelligent vehicle has a moving, diagnostic, pick up and delivery functions. Structural decomposition is based on the material components of agent, isolating groups of independent components. For example, a vehicle traction system can be isolated from it odometer. Modal decomposition can be seen in system whose elements posses several functioning modes. This can be illustrated by a vehicle with several modes to describe it state: ready, degraded, failure.

Once these we used these methods and isolated variables and process, we create, for this independent group, as many level as many expressed needs, in necessary resources to model these groups. These groups can contain heterogeneous agent if their representation possess similar needs. By this way, during the simulation an entity of the studied system, will be represented by agents situated on different levels.

This part of the article illustrate that some agents at different levels are not obligatory situated at different scales but at different level of representation, more or less detailed or supporting more or less functionality. The following figure illustrate that fact, showing three agents at different levels, obtained with functional decomposition, these agents can potentially represent the same vehicle. During the simulation a vehicle can be represented by an agent of one of the three levels according to the simulation needs.

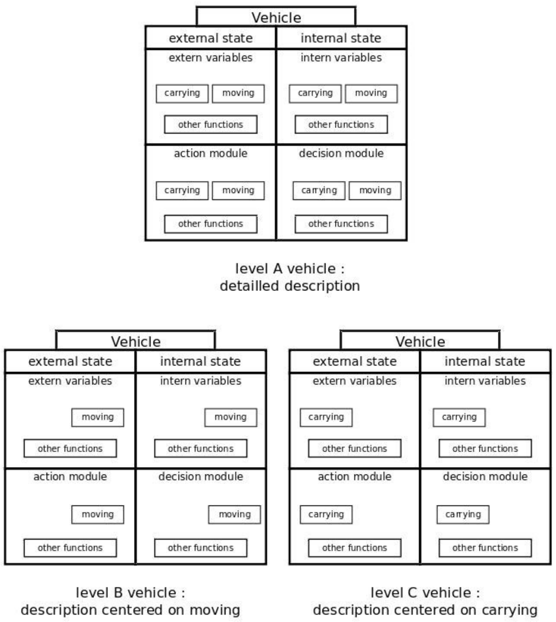

Figure 3: 3 Levels of Agents at the Same Scale

4.3. Relevant elements in a multi-level simulation

4.3.1. The organizations viewpoint

When agents of a single scale group themselves, it's possible to create in the simulation the formed organization and the corresponding agent at an smaller level of granularity. This is interesting to represent characteristics proper to the organization no deductible only from the agents member characteristics. Also, it permit to approximate or aggregate the life of the members and their interactions in the organization. Representation of some agents with bigger granularity is contained in the organization agent. That's make these member agents useless for a time and they can be delete of the simulation.

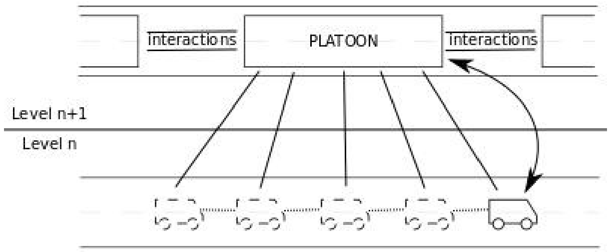

Figure 4: Organization Viewpoint driven implementation

The above figure illustrates this case. Five vehicles grouped themselves to form a platoon. That gaves birth to a platoon agent representing this group. Because the platoon agent can approximate the variables and the results of the communications between members of the platoon the member agents representing following vehicles are no more needed in the simulation and can be delete. The member agent representing the leader vehicle is needed because there is no function, in the platoon agent, which can approximate it routing. So it have to share it routing with the platoon agent (this interaction is represented by the double arrow).

4.3.2. The composing viewpoint

When modelers wish to represent some entities of the system at a given scale, it's no more necessary to keep the personification of groups formed by these entities because this induce a redundancy of data and process. This is illustrated by the next figure with the last shown example. The modelers wish to observe the platoon vehicles individually, that requires the existence of the agent vehicles in the simulation. Once these agents have been created the agents representing the groups they form (here a platoon) can be delete. At that moment the life of the organization and it interactions with others ones are supported by the vehicles agents.

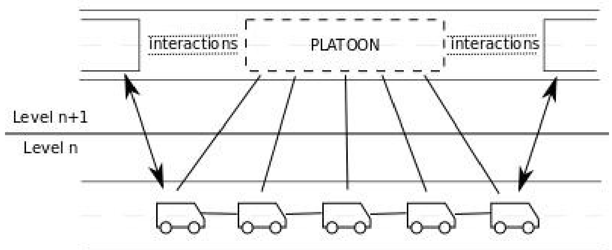

Figure 5: Members Viewpoint driven Implementation

4.4. Level temporality

We use the framework of temporalities simulation system (Zeigler 2000). No constraints is imposed on the scheduling mode (time or discrete events), but the fact that the scheduling is distributed between level. This distribution makes more sens than a synchronization on others “levels” like the agent one (Weyns 2003) or the system one (Michel and al. 2003), which are not adapted for our problematic.

Levels can have different temporal dynamics. Independently of the other levels, it's interesting to give a level a dynamic temporal whose time time step is higher as possible. This is make in order to update the dynamic state of this level as less often as possible, respecting the wish of the modeler. That is to say, giving to the agents the longest possible life cycle which stay coherent with the rest of the simulation. IRM4MLS is a structured interactions model. Morvan and al. (Morvan and al. 2010) propose an algorithm adapted to IRM4MLS which manage the coupling between levels with different temporal dynamics. That permits to apply easily the proposed methods above.

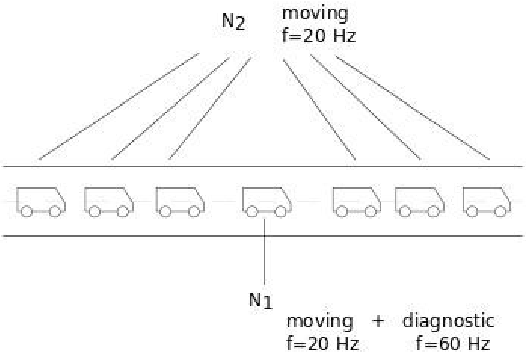

Figure 6: Example of Multi-Agent Multi-Level modelling with Different Temporalities

The preceding figure permit us to mention different constraints which fix the life cycle term of agents in a same level. Here we speak about the frequency of a level expressed in Hertz. This indicate how many times a second, it's necessary to execute the updating processes of the dynamic state of a level. Let's imagine that all functions of an agent possess a minimal frequency beyond which the simulation of this function is no more realistic. If a level permit to his agent to dispose of functions with different frequencies, it adopts the higher one, to keep a correct simulation of the functions with this frequency.

That's why in the example of the figure 6 the frequency of the level $N1$ is equal to 60z because the diagnostic function of the modeled vehicles needs this minimal frequency.

The other constraint come from the interactions between levels. If we continue with the preceding example, let say that t $N1$ he level needs a minimal frequency equal to 20 Hz, the modeler could allocate this frequency to $N2$. However if the $N1$ level is influenced by a $N2$ nd have to calculate the reaction induced by these influences at a frequency higher to 20 Hz (logically less or equal to 60 Hz) it is possibly necessary to allocate a higher frequency to $N2$. So it is necessary to dynamically modify the frequency of a level N and adapt it to the changing needs (for example, when a level with a higher frequency receive influences from N is created) of the simulation and give back to his level his minimal frequency, defined during the implementation phase, when no needs are expressed.

5. CONCLUSION

In this article we presented a methodology permitting to give a bigger representation power to a multi-levels MAM representing a big number of agents which maintain complex multi-levels interactions.

It's necessary to start by decomposing the agents at a same given scale and for each obtained representation we allocate to it at a different level. Thus in function of needs a same entity pass from a level to another one. After that, we define the elements which can be simplified and supported by organizations and conversely which elements of the organization can be supported by its members. At last, it 's convenient to fix the minimal frequency for each level, defined before, specifying which frequency have to adopt a level when it interacts with another one according to the interactions between levels.

However in this article we do not describe in detail the mechanisms to approximate, aggregate or stock outside of the simulation data, at some times of the execution. Also we do not evoke the problem which can be the loss of significant information related to these mechanisms, neither get round them. To test the global coherence of our simulation we should measure the consistency of it, as expressed in (Davis et al. 1993).

Our future work will consist in creating a framework adapted to IRM4MLS, with a strong formalism, which permits dynamic changes of level of detail during the simulation.

REFERENCES

Angelo, L., 2010. Multi-scale methods and complex processes: A survey and look ahead. *Computer and Chemical Engineering* 34 : 1467-1476.

Calvez, J.-P., 1990. Spécification et conception des systèmes : une méthodologie. Editions Masson, ISBN 2-225-82107-0.

Davis, P. and Hilestad, R., 1993. Falilies of Models that Cross Levels of Resolution : Issues for Designe, Calibration and Management. *In proceeding of the 25th Winter Simulation Conference (WSC'93)*.

Drogoul, A, Vanbergue, D, Meurisse, T, 2002. Multi-Agent Based Simulation: Where are the Agents. *Proceedings of MABS.*

Ferber, J, 1995. *Les Systèmes Multi-Agents: Vers une Intelligence Collective.* InterEditions.

Ferber, J, Müller, J-P., 1996. Influences and reaction: a model of situated multiagent systems. In *2nd International Conference on Multi-agent systems (ICMAS-96)*, pages 72–79.

Gil Quijano, J, Hutzler, G, Louail, T, 2009. De la cellule biologique à la cellule urbaine : retour sur trois expériences de modélisation multi-échelles à base d'agents. In *Actes des 17èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09)*.

Michel, F, Gouaïch, A, Ferber, J, 2003. Weak interaction and strong interaction in agent based simulations. *Lecture Notes in Computer Science*, 2927 :43-56,.

Michel, F, 2007. Le modèle irm4s. de l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multiagents. *Revue d'Intelligence Artificielle*, 21 : 757–779.

Morvan, G, Veremme, A, Dupont, D, Jolly, D, Charabidze, D, 2008. Vers une modélisation multi-niveaux. *Acte de la 7ème conférence de Modélisation et Simulation MOSIM*, Paris, France, 1, 167-174.

Morvan, G, Veremme, A, Dupont, D, Jolly, D, 2009. Modélisation et conception multiniveau de systèmes complexes : stratégie d'agentification des organisations. *Journal Européen des Systèmes Automatisés*, 43 :381–406, 2009.

Morvan, G, Veremme, A, Dupont, D, 2010. Irm4mls: the influence reaction model for multi-level simulation. In *11th International Workshop on Multi-Agent-Based Simulation*.

Navarro, L., Flacher, F. and Corruble, V., 2011. Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations. *Proc. of the 10th Int. Conf. On Autonomous Agents and Mutiagent Systems (AAMAS 2011),* 701-708, May, 2-6, 2011, Taipei, Taiwan.

Servat, D, Pierrer, E, Treuil, J-P, Drogoul, A, 1998. Towards Virtual Experiment Laboratories: How Multi-Agent Simulations Can Cope with Multiple Scales of Analysis and Viewpoints, pages 205–217. *Springer-Verlag* Berlin Heidelberg.

Pumain, D, Sanders, L, Bretagnolle, A, Glisse, B, Mathian, H, 2009. West The Future of Urban Systems: Exploratory Models. In: Lane, D, Pumain, D, Van der Leeuw, S, West, G, eds. *Complexity Perspectives in Innovation and Social Change.* Netherlands : Springer, 331-360.

Soyez, J-B, Morvan, G, Merzouki, R, Dupont, D, Kubiak, P, 2011. Modélisation et simulation multi-agents multi-niveaux. *Studia Informatica Universalis.*

Van Brussel, H , Wyns, J, Valckenaers, P, Bongaerts, L, Peeters, P, 1998. Reference architecture for holonic manufacturing systems: PROSA *Computers in Industry* 37, 255-274.

Weyns, D, Holvoet, T, 2003. Multi-Agent System Technologies, chapter Model for Simultaneous Actions in Situated Multi-agent Systems. Number 2831 in: *Lecture Notes in Articial Intelligence*. Berlin Heidelberg: Springer-Verlag, 105-118.

Zeigler, BP, Kim, TG, Praehofer, H, 2000. Theory of Modeling and Simulation. *Academic Press*, 2nd edition.