

A Multi-level Model for Multi-agent based Simulation

Thomas Huraux^{1,3}, Nicolas Sabouret² and Yvon Haradji¹

¹EDF R&D, Clamart, France

²LIMSI-CNRS, University of Paris-Sud, Orsay, France

³LIP6 - Pierre and Marie Curie University, Paris, France

Keywords: Multi-level Modeling, Multi-agent Systems, Simulation.

Abstract: In this paper, we consider the problem of modeling complex systems at several levels of abstraction. We design SIMLAB, a multi-level model for multi-agent based simulation. Our approach is based on the coexistence of different levels during simulation to enhance the model with complementary experts' opinion. We present how a same concept can be defined independently of its granularity using the notion of modeling axis. We consider recursive agents with interactions and influences which captures the inter-level dynamics. We also propose observations to detect and to reify macroscopic entities.

1 INTRODUCTION

The simulation of complex phenomena is based on the use of models that allow experts to understand systems and to anticipate changes. These models are highly constrained by the modeler's view of the system and by the computational limits, in terms of memory or computation time required to run simulations. However, we know that considering a complex system as a whole to reproduce the immense diversity of its behavior is unfeasible (Batty and Torrens, 2001).

This is the reason why simulations are based on partial representations. Rather than considering the whole picture, people use simplifications and abandon some aspects that they consider less relevant for a given study. The role of experts in the modeling process is then to make choices about what should be considered in the model.

Multi-agent systems (MAS) provide a well-suited paradigm for simulating such complex phenomena. One of the main strength of multi-agent simulation is that it uses distinct entities to represent every concept that has been identified as important by the modeler. Moreover, the behavior of every agent, *i.e.* every significant entity to the modeler, can be recorded and analyzed. As a consequence, and in contrast to other approaches, the produced model does not take the form of a "black box": as Edmonds explains (Edmonds, 2001), MAS offer an ideal support to interact with experts based on the agent's behavior analysis.

Another advantage of multi-agent based simula-

tion is that it can use multi-level organisation. Indeed, for a given complex system, expert knowledge is available at different granularity levels, depending on the domain, based on the idea that expert knowledge is largely unusable if we just consider the microscopic level (Conte et al., 2001). For instance in social science, studies go from psychology (individuals) to sociology (groups) through ergonomics (activity). In multi-level agent-based simulation, entities that correspond to different abstractions' level will be represented by agents at different levels in the MAS organization.

However, in most existing work, the analysis of the emergent phenomena in multi-agent systems only considers expert knowledge from one level at a time. In this paper, we claim that some behaviors cannot be understood without a multi-level context, *i.e.* a positioning in what forms the agent (entities at the lower levels) and what constrains it (entities from upper levels). Hence, our hypothesis is that representing and co-simulating different levels in the same model may produce agents with more complete behaviors.

The work presented here relies on two core ideas. First, common characteristics of the same concept shall be defined independently from its level of abstraction. Second, levels should coexist during the simulation not as different visualizations of a single phenomenon, but to connect different aspects, different expert knowledge. As we explain in the section 2, most of the existing models were not designed to co-simulate complementary points of view. They are

designed to perform abstractions or aggregations that control the microscopic level. In section 3, we present our multi-level model. Then, we illustrate in section 4 its properties with a simple example and in section 5 we discuss the perspectives of our works.

2 POSITIONING

Several frameworks and methodologies have been developed for engineering multi-agent systems. For instance, the MaSE (Wood and DeLoach, 2001) methodology proposes an iterative progression to build a multi-agent system. The methodology is divided into six phases but none of them is the question of choosing levels of abstraction, even less a possible coexistence of these levels. Yet we believe that it should play a central role in the design process of systems. In Gaia (Zambonelli et al., 2001), AGRE (Ferber et al., 2005) or INGENIAS (Pavón and Gómez-Sanz, 2003), the use of organizations can be considered as an addition in the system of other levels of abstraction. In these works, organizations are groups of agents with tasks or/and goals in common. Agents are then subordinated to organizations in which they belong and this organization strongly relies on the agents' goals. In our work, we propose a multi-level model for simulation with intentional agents without explicit goals manipulation.

Multi-level models in the literature are often domain-dependent. For instance the RIVAGE model in (Servat et al., 1998) which consider water as a multi-level set of agents for modeling flow, erosion and infiltration on heterogeneous soils. Another example : the model in (Tranouez, 2005) simulate liquid flow where authors use vortex and macroscopic-vortex models. These specific models cannot be reused in a different context. But what is important with these models is that they are not designed to lower computational cost but to facilitate studies using several levels. This idea is at the heart of our work. Actually, it is not one of our objectives in this current paper to build a generic model.

In contrast, other works use multi-level modeling to speed up simulations. For instance, the level-of-detail approach used by (Navarro et al., 2011) to realize large scale urban simulation. The system selects by itself the level of representation for each agent allowing a significant computational gain. In such systems, only one level is activated at the same time and so several levels do not coexist. We notice the same problem with the hierarchical multi-level model SWARM (Minar et al., 1996) and the SimulBogota model (Gil-Quijano et al., 2007) based on agent ag-

gregation. In these models, macroscopic entities take the lead of microscopic ones which loose their autonomy. Here, even though macroscopic levels are explicitly represented, levels do not really coexist.

In (Nguyen et al., 2011), authors choose a multi-modeling approach using heterogeneous models to represent several levels. Agents are aggregated and constitute path using equations of fluid flow to simulate evacuation in case of tsunami alert of the vietnamese city Nha Trang. This approach is studied in (Siebert et al., 2010) : the meta-model AA4MM use artifact to couple models of several levels. In these models, inter-level interactions allow coupling between models with functions of information sharing. Although multi-modeling enable to join existing models from different domain, it can be presumably more interesting to get domain experts together to exchange looks on a same model.

The interested reader can obtain additional references in the literature review on multi-level agent-based modeling by G. Morvan (Morvan, 2012). Based on the literature, we identify two limits in the existing models :

1. Levels do not really coexist during the simulation with often controlled microscopic levels.
2. Interactions and perceptions between levels are not clearly elucidated.

For that reason we propose in this article a multi-level model for multi-agent simulations (SIMLAB stands for SIMLAB Is Multi-Level Agent Based) in which various level agents can coexist and their interactions reflect the properties of common notions between entities, as understood by the modeler.

3 THE SIMLAB MODEL

Our multi-level agent-based model is based on the following fundamental concepts. First, some characteristic elements of the modeled entities are shared by agents from different levels. To this purpose, we introduce the notion of *modeling axis*, which captures the representation of transverse agents' components. Second, we propose to distinguish between agent *interactions* (i.e. exchange of informations or requests to the purpose of the system's execution) and intra-axis *influence* of properties, which captures the dynamics of these transverse components. Last, in order to have a dynamic reorganization in the MAS, we propose to use *observations* that detect and reify macro entities that make sense for the experts.

The following subsections present the general agent model and these four components of our multi-

level MAS model.

3.1 Agent Model

All entities in the system are recursive agents (agents composed of other agent with properties and interactions transferred from one level to another). Let Ω be the set of all agents. For each agent $\omega \in \Omega$, we have Ω_{sup}^{ω} the set of direct super-agents of ω and Ω_{sub}^{ω} the set of direct sub-agents. We denote \sqsubset the non-transitive inter-level relation: $\omega_1 \sqsubset \omega_2$ means that $\omega_1 \in \Omega_{sub}^{\omega_2}$ and $\omega_2 \in \Omega_{sup}^{\omega_1}$. We assume that \sqsubset is acyclic, but no other restriction is specified in our model. In particular, an agent can have several super-agents.

Every agent $\omega \in \Omega$ is characterized by: a set of properties \mathcal{P} (see 3.1.1), a set of actions \mathcal{A} (see 3.1.2) and interactions I (see 3.1.3), and a set of observations \mathcal{V} (see 3.4).

3.1.1 Agent Properties

Agent properties are the variables that can be manipulated by the agent. They capture the characteristic notions of the modeled entity. A property can be atomic (real, boolean, ...) or more complex (list, set, or even another agent). We denote $\omega.p$ the property p of agent ω . We denote \mathcal{P} the set of all properties and $\mathcal{P}(\omega)$ the set of properties of ω .

3.1.2 Agent Actions

Each agent has a set of internal actions to modify its properties. To describe the effect of an action, we introduce the operator “:=” to change state properties : given a value $\omega.p := v$ means that v is allocated to the property p of the agent ω . We denote $\mathcal{A}(\omega)$ the set of actions of ω . Each $a \in \mathcal{A}(\omega)$ is a set of property allocations. Moreover, we associate a precondition $Pre(a)$ (i.e. a function of $\mathcal{P}(\omega)$ to \mathbb{B}) to each action a and, at every step of execution, an agent performs all its actions whose preconditions are satisfied.

3.1.3 Interactions between Agents

Our interaction mechanism is a simplified version of the classical intentional communication model proposed by FIPA (FIPA consortium, 2003).

Every agent $\omega \in \Omega$ is associated with a set of reactions $\mathcal{R}(\omega)$, i.e. actions that can only be triggered by interactions. Like actions, reactions are a set of properties allocations and are associated with a precondition ($Pre(r)$).

Moreover, every agent ω has a set of interactions $I(\omega)$, where each $i \in I(\omega)$ is a tuple

$\langle target, reactions \rangle$ with $target \in \Omega$ the recipient and $reactions \subseteq \mathcal{R}(target)$ a set of reactions by the recipient. As for reactions and internal actions, we associate a precondition $Pre(i)$ to each interaction.

At every step of execution, for all i such that $Pre(i)$ is satisfied, for all $r \in reactions(i)$ such that $Pre(r)$ is satisfied, all the effects of r are performed by $target$.

Example. Let n and h be two agents $\in \Omega$ corresponding to an individual and a heater, $h.temp \in \mathcal{P}(h)$ the temperature property of h and $increase \in I(n)$ the possible interaction of n on h define as :

$$increase = \langle h, \{h.temp := h.temp * 1.1\} \rangle$$

If the individual performs the *increase* interaction on the heater, it will increase the temperature by 10%.

3.1.4 (De)Activation of Agents

Each agent can be activated or de-activated during the simulation, either for selecting a suitable simulation's level for a given study, or for reducing the computation load. We note $\omega.active$ the property that states whether an agent is active or not, and we have two set of actions f_{on} and f_{off} which are triggered when the agent activates (resp. de-activates). When de-activated, an agent continues to influence its peer properties but it will no longer interact with other agents (this interaction task is moved to the responsibility of its super-agents) or perform internal actions.

3.2 Modeling Axis

A modeling axis is intended to represent consistency between super and sub-agents. It consists of a set of entities that correspond to the same concept. In other words, it regroups all the abstraction levels of a same aspect of the phenomenon being studied. For instance, individuals, families and social groups are elements of the population axis.

In our model, we have the to following properties:

- agents in the same axis are connected with the relation \sqsubset ,
- agents in the same axis share a set of common properties.

Notation. We denote $\chi_c \subseteq \Omega$ the modeling axis corresponding to concept c (e.g. $\chi_{population}$).

3.2.1 Agents Relation in an Axis

Every axis χ_c forms a connex subset of Ω for the relation \sqsubset : all agents $\omega_i \in \chi_c$ are connected to the others $\omega_j \in \chi_c$ via \sqsubset .

We denote $lvl(\omega)$ the agent's level in its axis, with $lvl(\omega) = 0$ for lower-level agents, and $lvl(\omega) = \max_{m \in \Omega_{sub}^{\omega}} (lvl(m))$ for higher-level agents. The modeler can activate or de-activate agents of a given level, for the purpose of its study or for performance issues. $\chi_c(i) \subseteq \chi_c$ represents the set of all agents with $lvl = i$.

3.2.2 Shared Properties in an Axis

It is important that different modeling levels formed by the relationship “ \sqsubset ” mean something to the modeler. Indeed, there is an infinite number of levels for a same phenomenon, and they are not all relevant in the context of a given study. What links these different levels within a modeling axis, conceptually, is sharing some properties. Consider the example of human activity, it can be modeled as a sequence of tasks, or at a higher level as a set of habits. With these two models of the same phenomenon, there are common properties like conditions, preferential periods, frequency of realization, etc.

Every axis χ_c is characterised by a subset of properties $\mathcal{P}^{\chi_c} \subset \mathcal{P}$ such that:

$$\forall \omega \in \chi_c, \forall p \in \mathcal{P}^{\chi_c}, p \in \mathcal{P}(\omega) \quad (1)$$

We call \mathcal{P}^{χ_c} the set of *shared properties* of the axis χ_c .

3.3 Intra-axis Influences

To describe inter-level relations in a modeling axis, we introduce the notion of influence. It is intended to represent how agent properties are influenced by its super or sub-agents. In MAS, we find the notion of influence in (Morvan et al., 2011) where the influence is the “desire” of an agent to modify its environment. The semantic is different here, we are closer to the definition of social influence and the influence of individuals on the group as it is found, for example, in work on social computing (Wang et al., 2007). Through this mechanism, agents of different levels will be able to co-evolve during the simulation.

In the model, we define \mathcal{F} as the set of influences. An influence $f \in \mathcal{F}$ is characterized by a tuple $\langle \omega_e, \omega_r, P_{src}, p, infl \rangle$. It allows an agent ω_e to influence the value of a property of one of its super or sub-agents called ω_r , based on some of its properties. $P_{src} \subseteq \mathcal{P}(\omega_e)$ is the set of properties of ω_e used to compute the influence on ω_r , $p \in \mathcal{P}(\omega_r)$ the changed property and $infl$ the influence function which change the property. We denote $f(P_{src})$ the value that ω_r must integrate in p (with the relation $:=$).

Example. Let n be a sub-agent of f . They correspond to an individual and its family. $t \in [0, 1]$ is an

individual's property which represents its tendency to increase the heater. $prio \in \{co, sp\}$ describes if the family gives priority to its comfort or its spending. We define $\in \mathcal{F}$ the following influence:

$$\langle f, n, prio, t, \left\{ \begin{array}{l} t * 0.5 \quad prio = sp \\ t \quad prio = co \end{array} \right\} \rangle$$

It means that the individual's tendency to increase the heater is halved when its family give priority to spending.

When an agent is de-activated, it continues to have an influence, unlike the actions and interactions that the agent can not achieve anymore. The idea is that the agent's behavior is always potentially influenced by the presence in the system of its super and sub-agents, regardless of their activation status.

Note that influences apply on properties, not agent instances of properties. As a consequence, shared properties have the same influences for all agents in the axis.

Remark. Considering the influences as a directed graph with \mathcal{P} the set of nodes (corresponding to the set of properties, potentially involved in influences), we must constrain the model to prevent the presence of cycle. Let $(p_0, p_1, \dots, p_{n-1}, p_n)$ be a sequence of nodes in which two consecutive nodes p_i and p_{i+1} are connected by an influence. The system must verify the following acyclic property :

$$\nexists (p_0, p_1, \dots, p_{n-1}, p_n) \mid p_0 = p_n \quad (2)$$

3.3.1 Influences Integration

Each agent property will be influenced by other properties in the same modeling axis. The agent is responsible for aggregating all these influences to determine the value of its property, using a σ combination function (which can be a sum, a product or any combination chosen by the modeler) :

$$\omega.p := \sigma(E(p)) \quad (3)$$

with $E(p)$ representing the set of all influences.

3.3.2 Recursive Properties

The concept of influence allows us to define recursive properties. A recursive property is a property that is defined for all agents of an axis (*n.b.* recursive properties are also shared properties) and is influenced by all the source property values in the lower-level agents. The influence of a recursive property is always oriented from the sub-agent to the super-agent.

The introduction of shared properties and recursive properties in the model, with the influence's mechanism, allows the modeler to establish consistency in behavior between different levels (*e.g.* establishing a link between the preferences of an individual and those of a group he belongs). On one side the shared properties constitute the structural elements of a modeling axis. On the other, the influence and recursive properties link levels together and form a common dynamic.

We denote $\mathcal{P}_{rec}^{\chi_c} \subseteq \mathcal{P}^{\chi_c}$ the sub-set of recursive properties of the χ_c axis. For each shared property p between an agent a and its sub-agents Ω_{sub}^a , there is a corresponding influence. Formally :

$$\forall p \in \mathcal{P}_{rec}^{\chi_c} \quad \forall i > 0 \quad \forall a \in \chi_c(i) \quad \forall \omega_e \in \Omega_{sub}^a \\ \langle \omega_e, a, P_{src}, p, infl \rangle \in \mathcal{F} \quad (4)$$

3.3.3 Influence of a Recursive Property

Changing the value of a recursive property translates into a recursive function whose computation is from the agent to the sub-agents. Indeed, an agent $\omega \in \Omega$ recursively computes the influence f (see 3.3) for each of its properties, such as :

$$f(\omega) = \begin{cases} \omega.\sigma(f(sub_1), \dots, f(sub_n)) & \Omega_{sub}^\omega \neq \emptyset \\ 0 & \Omega_{sub}^\omega = \emptyset \end{cases} \quad (5)$$

with $sub_i \in \Omega_{sub}^\omega$, σ the combination function and 0 the neutral element for the influenced property.

3.4 Self-observation and Transformations

We consider that to be effective, our approach to multi-level modeling requires a dynamic structure. Agents should be able to reorganize themselves in response to the entrance or exit of agents, to allow dynamic formation of super-agents, or with an eye to reify emergent phenomena. To this end, each agent can be associated with one or more observations represented as a tuple $\langle S, M, \phi, t \rangle$ with $S \subseteq \Omega$ the measured agents. Each measurement is then compared (using the following function : $M : S \rightarrow \mathbb{R}$) with an activation threshold $\phi \in \mathbb{R}$ to potentially trigger a transformation $t \in \mathcal{T}$ (see 3.4.1).

3.4.1 Transformation of the System

A transformation modifies the organizational structure of the system. In the multi-level representation domain, we find this type of operation with holonic systems based on Koestler's work (Koestler, 1967),

for instance the CRIO model in (Gaud, 2007). Let \mathcal{T} be the set of transformations. A transformation $t \in \mathcal{T}$ is a couple $\langle cond, \Delta^t \rangle$ with $cond$ a condition on the MAS structure for the completion of t and Δ^t a set of modifications on the organizational structure of the system.

When an agent triggers a transformation, it is always directed to a single target agent. Transformations are defined independently of agents: for a given transformation's type, there is potentially $|\Omega|^2$ transformations. Agents have five types of transformations:

- (1) **Create.** The target agent is added by to the set Ω .
- (2) **Join.** The target agent is added to the set of super-agents of the triggering agent.
- (3) **Merge.** The triggering agent and the target agent group together to form a new super-agent. This transformation is composed of a *create* transformation following by two *join*. Note that two agents cannot "merge" if they already have a common super-agent with the same properties: common super-agents at a given level must have different properties. However, two *individuals* could form both a *family* and a *group*.
- (4) **Leave.** The target agent is removed from the set of super-agents of the triggering agent.
- (5) **Delete.** The target agent is removed from the set of agents Ω .

3.4.2 Triggering a Transformation

When the measurement function M , applied to one of the agents of S , exceeds the threshold ϕ , the observation triggers the associated transformation $t \in \mathcal{T}$:

$$\exists x \in 2^S | M(x) \geq \phi \rightarrow t(x)$$

4 ILLUSTRATIVE EXAMPLE

To illustrate our proposal, let us provide a simple example of multi-level modeling : we consider the thermal comfort of individuals and groups. Individuals in a room has a thermal comfort which will encourages them to adjust their heater according to the group.

Applying the proposed model we introduce two modeling axis : the population and the environment. The population is composed of individuals that can form groups and environment is limited to heater located in rooms. As shown in figure 1, each individual (which influence each other within the group) will interact with the heater to adjust the thermostat according to its thermal comfort. Then, the heater influences

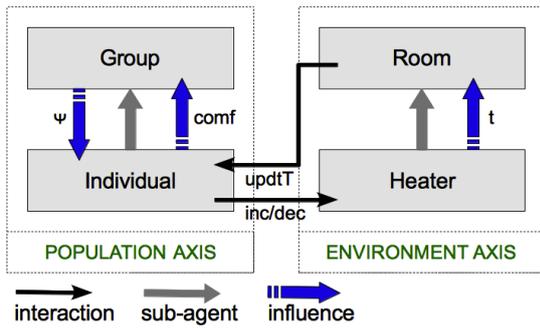


Figure 1: Overview diagram.

the room temperature. Finally the room interacts with the individual to change the perceived temperature.

4.1 Population Axis

We denote χ_{pop} this modeling axis. It contains all populations on several levels. We denote $comf \in \{-2, -1, 0, 1, 2\}$ the recursive property representing the level of thermal comfort. As shown in 3.3.2, it means that this property is influenced by all the source property values in the lower-level population agents.

4.1.1 Individual

At the microscopic level we have the individuals. An *individual* is characterized by $ct \in \{-1, 0, 1\}$ a cold-tolerance level, $pt \in \mathbb{N}$ the perceived temperature, $tend \in [0, 1]$ the tendency to adjust the heating, $\psi \in [0, 1]$ a probability to change it and $cr \in \chi_{env}$ the current room. The cold-tolerance changes the level of thermal comfort felt by individuals.

Action. An individual $i \in \chi_{pop}$ updates continuously its thermal comfort following :

$$i.comf := i.pt - (21 + i.ct)$$

For this example, we simply consider 21°C as the ideal temperature. This is an obviously caricatural description of thermal comfort mechanism but our example is intended to illustrate the possibilities of our model.

An individual modifies its tendency to adjust the heating according to group's thermal comfort.

Reaction. An individual modifies its perceived temperature accordingly to the room's interaction (see 4.2.2).

Interactions. An individual can act upon heater h with the following interactions:

- $inc = \langle h, \{h.th := h.th + 1\} \rangle$

- $dec = \langle h, \{h.th := h.th - 1\} \rangle$

The heater will influence the room's temperature which, in turn, impacts the individual's thermal comfort. A constant discomfort threshold is used as a precondition to trigger these interactions from the individuals.

Influence. Because thermal comfort is a recursive property, individual i influences the thermal comfort of the group g following :

$$\langle i, g, \{i.comf\}, g.comf, i.comf \rangle$$

Observation. We consider a simple observation which adds an individual i to a group when it enters into a room : $\langle Groups \subset \chi_{pop}, M, 1, join \rangle$ with

$$M : g \in Groups \rightarrow \begin{cases} 1 & g.room = i.cr \wedge i \notin \Omega_{sub}^g \\ 0 & \text{otherwise} \end{cases}$$

4.1.2 Group

A *group* is an agent composed of several individuals in the same room *room*.

Action. A group $g \in \chi_{pop}$ updates continuously its level of thermal comfort using the average function on influences to its comfort property (see σ in 3.3).

Influence. A group g influences the probability of its members to adjust the heating following :

$$\forall i \in \Omega_{sub}^g \langle g, i, \{g.comf\}, i.\psi, i.tend * \frac{|g.comf|}{2} \rangle$$

4.2 Environment Axis

Let χ_{env} be the environment axis. It contains the heaters and the rooms as seen as groups of heaters. Rooms are defined in a static manner, they do not evolve during the simulation.

4.2.1 Heater

A *heater* is characterized by a thermostat $th \in [0, 9]$.

Reaction. A heater modifies its thermostat accordingly to individuals' interactions.

Influence. A heater h influences the temperature of the room r with :

$$\langle h, r, h.th, r.t, r.t + (h.th * \alpha) \rangle$$

with α the heater's efficiency.

4.2.2 Room

We define a *room* as a super-agent of heater. It is characterized by a current temperature t .

Action. A room $r \in \chi_{env}$ updates continuously its temperature using an average function as σ on all the influences coming from heaters :

$$r.t = \overline{f(sub_1), \dots, f(sub_n)} \quad \forall sub_i \in \Omega_{sub}^r$$

We consider that rooms' temperature is only dependent on heaters' thermostat.

Interaction. A room r change the perceived temperature by individuals :

$updtT =$

$$\langle r, \{\forall i \in \Omega_{sub}^{group} \mid group.cr = r\}, \{i.pt := r.t\} \rangle$$

4.3 Discussion

This example shows how much the notion of modeling axis is implicitly present when we designing models. As it often happens with MABS, we have several concepts in this example. First, individuals and groups are entities which represent actors in the system. Second, heaters and rooms that represent the environment with our model. It is very straightforward to define the population and the environment axis corresponding to these concepts. As a result, we tend to obtain a multi-level representation which do not distort the modeled system. Moreover, shared properties facilitate the modeling task by encouraging the reflection around what is common to all levels in a modeling axis. Once this work is completed, there is no need to define these properties for each agent. Recursive properties extend this simplification of modeling to the dynamic's design. This is illustrated in the example : the thermal comfort is computed on the individual level and it is transmitted to the group. One can see how practical it could be for complex system modeling.

Interactions are at the heart of agent based modeling. However, it is not sufficient when entities of several levels are introduced into the model. This is why we introduce influences in addition to interactions. In fact, the difference is purely conceptual : On the one hand, interactions contribute to the system's dynamics (in this example, leading to a change of thermostat or thermal comfort). On the other hand, influences contribute to the coherence between levels. For instance, the comfort level of the group is always connected to the individuals' comfort. In fact, it would be possible

to represent influences using interaction (*e.g.* an individual interact with the group to change its comfort) but the main idea here is to separate them in order to simplify the modeling process. Thus, modeling strategy can then be summarized in four steps :

1. What are the concepts involved to define the corresponding modeling axis and then define levels for each of them ?
2. In which ways entities interact ?
3. How the behavior of agents influences their super-agent, and *vice versa* ?
4. What organizational elements should be dynamic and what could be the observations and the associated transformations ?

The balance between influences (*i.e.* non-autonomous variable modification) and interactions (*i.e.* autonomous exchange of information) need to be further studied, to decide how the system designer can have agent that are aware of other levels and more or less sensitive to influences. This is one of our current research objectives.

The role of observations as part of the organizational process, was not fully illustrated in our example. In order to better understand this feature, we must consider a more complete and complex example. Consider now that individuals can perform tasks (*i.e.* elements of everyday life activity : sleeping, eating, watching TV, ...), modifying their current room and their thermal comfort (some task such as cleaning leads to increasing the amount of body heat production and therefore changes the perceived temperature). Hence, third concept is introduced in our model leading to create a new modeling axis : the activity axis. Such an axis allows us to represent human activity on several levels of abstraction (*e.g.* tasks, habits, lifestyles, ...). With activities, we only need to provide observations to individuals such that they automatically form groups when they perform the same activity pattern. Groups can be interesting for the modeler in many studies. For instance in a study on energy consumption, groups are highly relevant as they can help domain experts to identify some typologies of consumers and their related energetic behavior. It must be emphasized that it is important that agents are able to perceive themselves and modify their organizations. Indeed, modeling is facilitated because some levels can be created automatically. Moreover, an explicit representation of reified entities gives them visibility in addition to a real role in the simulation due to possible influences.

5 CONCLUSIONS

In this paper we propose SIMLAB, a novel model for multi-level agent based simulation. This model has the particularity to explicitly define the notion of modeling axis allowing representation on others aspects than only actors. In addition, influences are designed to represent inter-level relationships and influences of recursive properties are spread from one level to another. Moreover, observations make the model capable of modifying the system's organization, as detecting and reifying macro-entities. As we have explained, it would be interesting to represent multi-level entities within a same model allowing experts of different domain to work together at the level that suits them best.

In the SMACH platform (Amouroux et al., 2013), we are currently working on the application of this multi-level model to simulate human behavior. The SMACH simulation platform is intended to study household activities and their relation with electrical consumption depending on specific pricing policies or appliance use. We are extending the SMACH model to study population, activity and environment at several levels of representation. Our goal is to evaluate possible incentives to diminish peak hours electricity demand. We may be able to evaluate our model in a three axis representation of human activity. For instance, such model may well be able to reproduce interesting social phenomena without having to simulate thousands of agents introducing explicit social entities.

REFERENCES

- Amouroux, E., Huraux, T., Sempé, F., Sabouret, N., and Haradji, Y. (2013). Simulating human activities to investigate household energy consumption. *Proc. of ICAART*, 13.
- Batty, M. and Torrens, P. M. (2001). Modeling complexity: the limits to prediction. *Cybergeo: European Journal of Geography*.
- Conte, R., Edmonds, B., Moss, S., and Sawyer, R. K. (2001). Sociology and social theory in agent based social simulation: A symposium. *Computational & Mathematical Organization Theory*, 7(3):183–205.
- Edmonds, B. (2001). The use of models-making mabs more informative. In *Multi-agent-based simulation*, pages 15–32. Springer.
- Ferber, J., Michel, F., and Báez, J. (2005). Agre: Integrating environments with organizations. In *Environments for multi-agent systems*, pages 48–56. Springer.
- FIPA consortium (2003). FIPA Communicative Act Library Specification and FIPA ACL Message Structure Specification. Technical report.
- Gaud, N. (2007). *Systèmes Multi-Agents Holoniques : De l'Analyse à l'Implantation*. PhD thesis, Université de Technologie de Belfort-Montbéliard.
- Gil-Quijano, J., Piron, M., and Drogoul, A. (2007). *Mechanisms of automated formation and evolution of social-groups : A multi-agent system to model the intra-urban mobilities of Bogota city*. Idea Group Inc.
- Koestler, A. (1967). *The Ghost in the Machine*. Hutchinson & Co.
- Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The swarm simulation system : a toolkit for building multi-agent simulations. *GEMAS Studies in Social Analysis*, Working Paper 96-06-042.
- Morvan, G. (2012). Multi-level agent-based modeling-bibliography. Technical report, LGI2A, Univ. Artois, France.
- Morvan, G., Veremme, A., and Dupont, D. (2011). Irm4mls: the influence reaction model for multi-level simulation. In *Multi-Agent-Based Simulation XI*, pages 16–27. Springer.
- Navarro, L., Flacher, F., and Corruble, V. (2011). Dynamic level of detail for large scale agent-based urban simulations. *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 701–708.
- Nguyen, T. N. A., Zucker, J.-D., Du, N. H., Drogoul, A., and Vo, D.-A. (2011). An hybrid equation-based and agent-based modeling of crowd evacuation on road network. *International Conference on Complex Systems*.
- Pavón, J. and Gómez-Sanz, J. (2003). Agent oriented software engineering with ingenias. In *Multi-Agent Systems and Applications III*, pages 394–403. Springer.
- Servat, D., Perrier, E., Treuil, J.-P., and Drogoul, A. (1998). When agents emerge from agents : Introducing multi-scale viewpoints in multi-agent simulations. *LNCS*, 1534:183–198.
- Siebert, J., Ciarletta, L., and Chevrier, V. (2010). Agents and artefacts for multiple models co-evolution: building complex system simulation as a set of interacting models. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 509–516. International Foundation for Autonomous Agents and Multiagent Systems.
- Tranouez, P. (2005). *Contribution à la modélisation et à la prise en compte informatique de niveaux de description multiples*. PhD thesis, Université du Havre.
- Wang, F.-Y., Carley, K. M., Zeng, D., and Mao, W. (2007). Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79–83.
- Wood, M. F. and DeLoach, S. A. (2001). An overview of the multiagent systems engineering methodology. In *Agent-Oriented Software Engineering*, pages 207–221. Springer.
- Zambonelli, F., Jennings, N. R., and Wooldridge, M. (2001). Organisational abstractions for the analysis and design of multi-agent systems. In *Agent-Oriented Software Engineering*, pages 235–251. Springer.