Multi-level Modeling as a Society of Interacting Models

Benjamin Camus, Christine Bourjot, Vincent Chevrier

To cite this version:

Benjamin Camus, Christine Bourjot, Vincent Chevrier. Multi-level Modeling as a Society of Interacting Models. [Technical Report] 2013. hal-00913038

HAL Id: hal-00913038
https://hal.inria.fr/hal-00913038
Submitted on 3 Dec 2013

Multi-level Modeling as a Society of Interacting Models

Benjamin Camus, Christine Bourjot, Vincent Chevrier
Université de Lorraine, CNRS, LORIA UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France.
INRIA, Villers-lès-Nancy, F-54600, France.
benjamin.camus@loria.fr christine.bourjot@loria.fr vincent.chevrier@loria.fr

Abstract

We propose to consider a multi-level representation from a multi-modeling point of view. We define a framework to better specify the concepts used in multi-level modeling and their relationships. This framework is implemented through the AA4MM meta-model, which benefits from a middleware layer. This meta-model uses the multi-agent paradigm to consider a multi-model as a society of interacting models. We extend this meta-model to consider multi-level modeling and present a proof of concept of a collective motion example where we show the ability of this approach to rapidly change from one pattern of interaction to another one by reusing some of the meta-model's components.

Keywords: multi-level, multi-model, multi-agent, collective motion.

1. INTRODUCTION

In this article, we are interested in the design of complex systems. Simulation is an important tool for the study of such systems because it allows testing different alternatives and different scenarios while limiting experimentation costs.

Complex systems naturally involve at least two levels of representation: local and collective. The former corresponds to the micro level while the latter corresponds to the macro level of the phenomenon.

Let us consider the example of pedestrians walking in a street. At the micro level, each of them moves individually, according to his/her personal goal, but the trajectory he/she takes is influenced by other pedestrians' behaviors. When these pedestrians evolve and interact at micro level, they can form a collective motion phenomenon of a crowd at macro level. This macro structure can be considered in its own with its proper attributes (common goal, average density, speed) and behaviors (Musse 1997). In fact, when describing this phenomenon, one can consider both levels: the micro level (pedestrians walking in the street) or the macro level (crowd or groups formation, extinction, evolution and interaction between macro level entities).

In order to study such phenomena (with individual and collective representations), multi-level modeling is a good approach as it explicitly represents these different levels. In this perspective, the different levels can coexist and influence each other.

This multi-level approach could be used, for instance, when there is a lack of expressiveness of one level and a second one is needed; when available data explicitly refers to different levels of representation; or when the modeling question is explicitly to study the mutual influences between levels whose dynamics have to be coupled.

Each level of description may correspond to a model. In that case, we propose to consider a multi-level representation with a multi-modeling point of view. The question becomes how to integrate these models, which may describe the phenomenon at different spatial and/or temporal scales and possibly with different formalisms. Such integration isn't trivial as it underlies issues about the consistency of the resulting representation like: How to manage the different formalisms? How to have a sufficiently rigorous multi-modeling process to guarantee some validity of the obtained multi-model? In case where the models are already implemented in simulators the question is also: how to reuse the models with their simulators and make them interoperate?

The AA4MM (Agents and Artifact for Multi Modeling) meta-model (Siebert 2010b) answers to this integration issue of heterogeneous models and their simulators. It relies on a multi-agent perspective where each model is seen as an agent and the interactions between agents correspond to data exchanges between models. The multi-model is seen as a multi-agent system (MAS). Using a meta-model is a mean to bring some rigor in the modeling process by adding constraints in the multi-model's design that prevent ill-formed models (in that sense our approach can be related to FURM (Ropella 2005)) and to allow automating the generation and evaluation of a multi-model.

In the next part, we present different facets of multi-level modeling that involve different kind of interactions between levels and that corresponds to different modeling objectives. Based on similarities between them, we propose a generic framework for multi-level dynamics coupling. We detail the implementation of this framework in the AA4MM meta-model. As multi-level representation were not included in AA4MM's original specifications we explain how AA4MM is extended to multi-level modeling with constraints specific to that kind of representation. Finally, we present our approach's advantages through the implementation of variations of a multi-level modeled collective motion phenomenon.

2. MULTI-LEVEL REPRESENTATIONS

In this section, we present different approaches (Section 2.1) and different examples (Section 2.2) of multi-level representations in order to show the different facets of the multi-level modeling. We present then a framework describing in a generic way some of these approaches.

2.1 Different approaches of multi-level modeling

Different approaches of multi-level modeling exist depending on the relation between the levels and the level-changing policies. Each of them answers to different representational need.

Multi-resolution modeling (MRM) is a model, or a family of models managing several levels of resolution (Davis 1993). A level is said to be at low resolution when it represents the phenomenon with a coarse description (with respect to a high resolution level).

Different kinds of MRM can be considered. In a selective viewing model, the simulation takes place only at the high-resolution level for precision need. The low-resolution level is obtained a posteriori for presentation or data analysis purpose. (Klir 2004).

In the aggregation-disaggregation approach, two kinds of entities are considered: the High Resolution Entity (HRE), which corresponds to a micro representation of the phenomenon and the Low Resolution Entity (LRE), which corresponds to a macro representation. Here an entity is only represented at one level of resolution at a time. Aggregation (from HRE to LRE) and Disaggregation (from LRE to HRE) operations are used in order to change of level of resolution. The policy for level changing is determined by the separation of the global modeled space into different areas called play-box. Entities inside a play-box are all represented at the same level of resolution. Aggregation-Disaggregation operations are also used to put entities interacting together at the same levels of resolution.

In a concurrent representation model, the different levels of resolution are concurrently maintained and can influence each other. Here an entity is simultaneously represented at different levels and is called a Multi-Resolution Entity (MRE) (Natrajan 1995). The MRE uses a mapping function in order to translate from one level to another (Reynolds 1997). The interest of such approach is that it is possible to study the mutual influences between levels.

In a holonic multi-agent system (Rodriguez 2007), agents organize themselves into holons that can then group themself into holon (a super-holon) according to pre-defined criteria that can be based for instance on simulation performance (Demange 2012). The advantage of such approach is that, despite the previous approaches represent the relationships between different levels of resolution, the holonic approach focuses on the formation/separation of higher-level structures.

2.2 Some examples of multi-level modeling

In the simulation of the evacuation of Nha Trang City (Vo 2012), the major issue is to deal with a huge number of citizen agents evolving on the roads, which considerably slows down the simulation. To increase simulation's performance, roads are divided into two kinds of patches: crossroad patches, and segments of the road patches. In the former, citizen agents are simulated individually while the later used a macro model of pedestrian flow in order to simulate their moves. Each patches is represented by a road agent. An organizational meta-model for multi-scale modeling is used in order to ensure transitions between patches. Road agents use “capture” and “release” operations on citizen agents. A road agent manages execution of all of its captured citizen agents. A captured citizen doesn’t have an autonomous behavior anymore. When a citizen agent is released by a road agent, it recovers its autonomous behavior. A road agent captures citizens when they reach the segments of the road patch’s borders. Then, captured citizens are considered as pedestrians in the macro model of pedestrian flow. Road patches determine the movements of captured citizens according to the macro model, and releases them when they exit patches’ borders.

In the 3D real-time multi-agent pedestrians simulation of (Gaud 2008), the major issue is to deal with computation cost in order to ensure an acceptable performance for the 3D visualization of the simulation while managing a large number of agents. The solution is to dynamically change the level of description in order to have the best trade-off between the accuracy of the simulation and the computation performance. A holonic model is used in order to organize and dynamically change the level of description. Then, in a low level of the holarchy, perceptions and actions of pedestrian holons are aggregated in a pedestrian super holon. Physics based indicators are used in order to determine whether pedestrians should be grouped or considered individually.

In SIMPOP3 (Gil-Quijano 2010), a micro model SIMPOPNano, describing the evolution of a city, is coupled with a mesoscopic model, SIMPOP2, describing the evolution of a system of cities in order to study their mutual interactions. The challenge here is to ensure an ontological translation between levels: the concepts of urban function (from macro to micro) and spatial efficiency (from micro to macro) are used to ensure the passage from a level to another.

In simulBogota (Gil-Quijano 2010), the challenge is to model the evolution of the city of Bogotá when having a lack of information on the behaviors of basic entities (household and housing). In order to deal with this issue, mesoscopic entities are considered (household and housing groups) with an arbitrary defined behavior. Then, groups are dynamically detected and reified at mesoscopic level at each simulation step using an automated classification mechanism of the micro entities. When reified, household groups share housing with an auction mechanism and the micro population evolves based on global rules.

In (El Hmam 2006), the hybrid traffic flow approach is to jointly use macroscopic and microscopic models of a traffic flow in order to take advantage of each kind of representation. The space is divided into several areas that are not modeled at the same level. A macro model describing in a global and aggregated way the phenomenon is used for areas that don’t necessitate a high level of detail. At the opposite, areas where the traffic flow is more subject to abrupt changes are represented by a micro (multi-agent) model. The challenge here is to ensure transition between models that are not of the same nature, between agents and aggregated values. An upward agent (from micro to macro) and a downward one (from macro to micro) are defined to fill these roles.

All these examples use two models to describe the same phenomenon at different levels of representation (micro and macro) and mapping functions to translate a state from one level to another (Natrajan 1995). They can be related to the problem of multi-perspective modeling where each sub-model of the multi-model represents a perspective of the phenomenon (here a macro and a micro perspective) (Yilmaz 2007). In order to reconcile these different perspectives a bridge model can be used (Seck 2012). This bridge model links the different perspectives as it represents how the modeled entities are related in the different models. Therefore, the bridge model can be in charge of the interactions between these perspectives as it makes conceptual translations between them by transforming the exchanged data between the models.

However, the nature of relationships between levels don’t convey the same constraints. The approaches or examples have to be considered separately according to the kind of representation of one state of the phenomenon. Some of them (concurrent representation, selective viewing) represent simultaneously the same states at two levels of representation whereas others (aggregation-disaggregation, holonic) consider only one level of representation at a time.

2.3 A framework for multi-level modeling

The framework we propose is a rephrasing of (Bourgine 2008). It makes explicit the relationships between the two levels (micro and macro). This framework explicitly represents all features common to the multi-level modeling cases we described in previous sections. In this article we specify how to implement it in order to consider simultaneously two levels of representation.

When representing an upward relationship (e.g. pedestrians are forming groups), we consider the following elements (see Figure 1):

- $\boldsymbol{X}_{t}$ is the set of information describing the state of the target system at time t at micro level.
- $\boldsymbol{Y}_{t'}$ is the set of information describing the state of the target system at time t' at macro level. As the macro level is a coarser description of the target system than the micro one, we can state:
  (1) $\mathbf{card}(\boldsymbol{Y}_{t'}) < \mathbf{card}(\boldsymbol{X}_{t})$**.**
- $\boldsymbol{f}$ is a function describing the micro dynamics. It specifies how to pass from X_t to X_{t+dt}.
- $\boldsymbol{g}$ is a function describing the macro dynamics. It specifies how to pass from $Y_{t'}$ to $Y_{t'+dt'}$.
- $\boldsymbol{u}$ is an upward function of integration of X_t ensuring the passage to $Y_{t'}$. According to constraint (1), it does an aggregation of information describing the phenomenon. This function fills the role of a bridge model as it is a translator between two perspectives of a phenomenon (micro and macro views).

Time representations at macro and micro levels are respectively t and t'. This means that the simulations may a priori have different execution policies or time steps. By using two notations, we explicitly ask the question of time correspondence between models. This means that $\boldsymbol{u}$ has also to manage the temporal mapping between levels.

$$\begin{tikzcd} Y_{t'} \arrow[r, "g"] & Y_{t'+dt'}=g(Y_{t'}) \\ X_t \arrow[u, "u"] \arrow[r, "f"] & X_{t+dt}=f(X_t) \arrow[u, "u"] \end{tikzcd}$$

Figure 1. Upward relationship between levels of representation and temporal evolution of the phenomenon.

When representing macro dynamics having an influence on the micro one (e.g. groups influence pedestrians), it is needed to add a new relationship (see Figure 2):

- $\boldsymbol{d}$ is a downward function ensuring the translation of $Y_{t'}$ in terms of influences on f, and f has to integrate this influence. Then the arity of f changes as this function takes two arguments: the micro states of the phenomenon (X_t) and the macro influences ($d(Y_{t'})$). According to the (1) constraint, d does an augmentation of information. As $\boldsymbol{u}$, $\boldsymbol{d}$ is also in charge of the temporal mapping between levels. This function can also be considered as a bridge model.

$$\begin{tikzcd} Y_{t'} \arrow[r, "g"] \arrow[dr, "d"] & Y_{t'+dt'}=g(Y_{t'}) \\ X_t \arrow[u, "u"] \arrow[r, "f"] & X_{t+dt}=f(X_t,d(Y_{t'})) \arrow[u, "u"] \end{tikzcd}$$

Figure 2. Upward and downward relationships between levels of representation and temporal evolution of the phenomenon.

This way of representing the different levels is sufficiently generic to be mapped with some of the cases of section 2.2. Indeed, the different elements are expressed in a generic way and then can describe several multi-level modeling example. Table 1 illustrates how this framework can be mapped on some of the previous cases.

Table 1. Description of some multi-level modeling cases through our framework.

Multi-level modeling cases	***X***	***Y***	***f***	***g***	***u***	***d***
Nha Trang City	Individual citizen in crossroad patches	Aggregated values	MAS	Equations of pedestrian flow	Capture operation	Release operation
SIMPOP3	City's districts and micro urban functions	Cities and macro urban functions	MAS. Micro urban functions localize selves in districts	MAS. Agents localize themselves in cities	Determine the spatial performance indicator based on district relationship	Determine cities' micro urban function attribute
SimulBogota	Household and housing	Household and housing groups	Population evolution rules	Auction mechanism	Classification mechanism	Relocation of household based on auction mechanism result

Moreover, as the framework is modular, it can be adaptated to different approaches of Section 2.1. (see Figure 3).

$$\begin{tikzcd}
Y_{t'} & Y_{t'+dt'} \\
X_t \arrow[u, "u"'] \arrow[r, "f"] & X_{t+dt} \arrow[u, "u"']
\end{tikzcd}$$

(a)

$$\begin{tikzcd}
Y_{t'} \arrow[dr, "d"] & Y_{t'+dt'} \\
X_t \arrow[u, "u"'] \arrow[r, "f"] & X_{t+dt} \arrow[u, "u"']
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
Y_{t'} \arrow[r, "g"] \arrow[dr, "d"] & Y_{t'+dt'} \\
X_t \arrow[u, "u"'] \arrow[r, "f"] & X_{t+dt} \arrow[u, "u"']
\end{tikzcd}$$

(c)

Figure 3. Different configurations of the framework. (a) A simple upward relationships corresponding to a selective viewing modeling. (b) A concurrent representation case (without macro dynamics) (c) A concurrent representation case with macro dynamics.

One can note here that our definition of multi-level coupling can't describe for example direct interactions between entities represented at different levels and approaches considering only one level of representation at a time. The models coupling doesn't have the same semantic for these cases and specifications have to be defined at the conceptual level in order to keep a rigorous modelling process.

If our framework seems to fit these cases in the realm of discourse, we have to investigate further with a concrete implementation. The next section describes how this framework can be instantiated with the AA4MM meta-model.

3. The AA4MM meta-model

3.1 The AA4MM's modeling paradigm

AA4MM (Siebert 2010b) considers a complex phenomenon as a set of interacting event-based models. It relies on the multi-agent paradigm: each model corresponds to an agent that is associated with its simulator, and the interactions are instantiated by artifacts (Ricci 2007). The global dynamics of the phenomenon is simulated thanks to the interactions between the models. Originality toward other multi-model approaches is to consider interactions in an indirect way, supported by an environment, and then to explicitly model the sharing of information between models.

The multi-agent paradigm richness of expression offers two advantages here:

- It is sufficiently flexible in term of representation, to be transposed to a multi-modeling point of view.
- The diversity of concepts (Agent & Artifact (Ricci 2007), Organization, Role (Ferber 1998)) is useful to convey the diversity of multi-modeling (Yilmaz 2004).

The AA4MM's modeling paradigm manages and links the two kinds of dynamics of a multi-model:

- The models' dynamics: It relies on the execution of the simulators implementing each model.
- The multi-model dynamics: the co-evolution of the different models. This dynamics is made explicit thanks to the behavior of the agents associated with each model.

This two dynamics organization allows reusing previously independent models while keeping the simulators they are implemented in, with only a small number of modifications to make.

This paradigm is formalized by the AA4MM's meta-model. The advantage of a meta-model for multi-modeling is that it allows directly manipulating the meta-modeling's concepts. These concepts have semantics that put constraints on the design process preventing the creation of ill-formed multi-model (Sprinkle 2010; Cetinkaya 2010). Moreover when these concepts have operational specifications, they can be then translated in executable software. This meta-modeling approach allows passing from the rigorous design of a multi-model to its implementation as a middleware with a clear separation of the different levels of question. In the next section, we detail the meta-models' concepts.

3.2 Concepts of the meta-model.

AA4MM relies on three concepts to describe a multi-model:

- The **m-agent** manages a model and is in charge of interactions of this model with the other ones (Symbol in figure 4a),
- Each of these interactions (between m-agents) is reified by a **coupling artifact** (Symbol in figure 4b),
- Finally, the **interface artifact** reifies interactions between a m-agent and its model (Symbol in figure 4c),

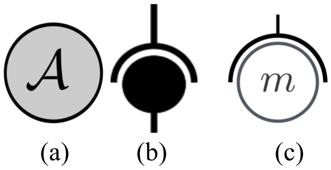


Figure 4. Symbols of the AA4MM components (a) m-agent, (b) coupling artifact, (c) interface artifact and its model m.

A m-agent's behavior corresponds to the cycle:

- **Read** data from the coupling artifact,
- **Update** the individual characteristics and execute one time step of its model via the interface artifact,
- **Write** data to the coupling artifact,

A coupling artifact proposes three functions :

- **write** : a m-agent managing a model m_1 writes data i_1 to be transferred. i_1 is a list of values, for example pedestrians' positions in a model,
- **transform** (scaling operation for example) : it changes i_1 data into transformed data i_1*,
- **read** : a m-agent reads transformed data i_1*,

The coupling artifact has a direction, its functioning is illustrated by Figure 5.

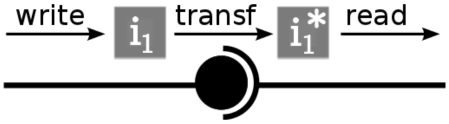


Figure 5. The three functions of the coupling artifacts.

An interface artifact contains primitives allowing manipulation of its simulator. It allows the following operations on the model: initialization, updating of the individual characteristics, execution for a time step and collecting information about individuals.

AA4MM relies on operational specifications and proved algorithms related to temporal coherence between models which allow implementing the multi-model and simulating it with only a selected number of functions specific to the application to define. Conceptual proofs of this approach with the coupling of Netlogo simulators have been done (Siebert 2011). We applied AA4MM to study the mutual influences between mobility models and routing algorithms in mobile ad-hoc networks (Leclerc 2010).

The original AA4MM's concepts have specifications for a structural coupling of models (Siebert 2010b). The structural coupling is defined as the evolution of a constant number of entities simultaneously involved in several dynamics, each dynamics being managed by a model. AA4MM manages the evolution of the entities' attributes.

In the following, we describe the implementation of a multi-model with AA4MM (Section 3.3) and how we extend the original specifications to implement multi-level coupling (Section 3.4).

3.3 Implementation of a multi-model in AA4MM

When implementing a multi-model with AA4MM we make the hypothesis that there are pre-existing simulators implementing each sub-model. In the following, we suppose a multi-model composed of two models m_1 and m_2 implemented respectively by two different simulators sim_1 and sim_2.

The preexisting AA4MM middleware consists of:

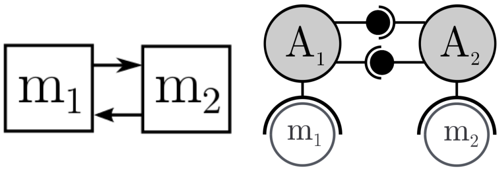
- The m-agent behavior corresponding to a proved multi-simulation algorithm.
- The coupling artifact.

In order to operationalize the multi-model, the interface artifacts have to be defined. Each simulator needs a specific interface artifact.

Features specific to the application have also to be defined:

- The data types of the application: the data exchanged between the m-agents (through the coupling artifacts) and the data to be collected from and injected in each simulator.
- Transformation functions between simulators (for example matching time and spatial dimensions). The proved algorithm of AA4MM needs these functions to ensure the consistency between the simulators' executions.

Figure 6 describes the coupling of two models (m_1 and m_2) implemented respectively with the simulators sim_1 and sim_2. Starting with an interaction graph between models (Figure 6a), a first step is to build its equivalent with the AA4MM meta-model (Figure 6b).



(a) **(b)**

Figure 6. (a) Interaction graph between models m_1 and m_2, (b) AA4MM corresponding meta-model diagram.

The meta-model implementation starts from existing elements: the AA4MM middleware and the two simulators (Figure 7).

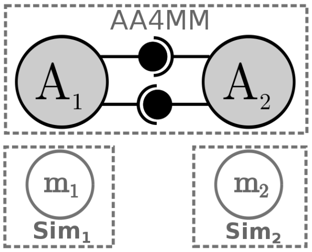


Figure 7. The starting components: the models (here m_1 and m_2) implemented in their simulators (here sim_1 and sim_2), the m-agents and the coupling artifacts.

They have to be completed with an interface artifact for each simulation software, with data types and dimensions matching functions between models (Figure 8).

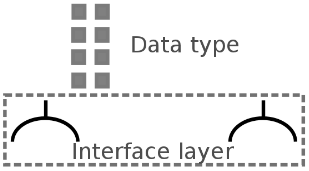


Figure 8. Features specific to the application: data types and the interface artifacts specific to a simulator.

Figure 9 shows the complete implementation of a multi-model with AA4MM.

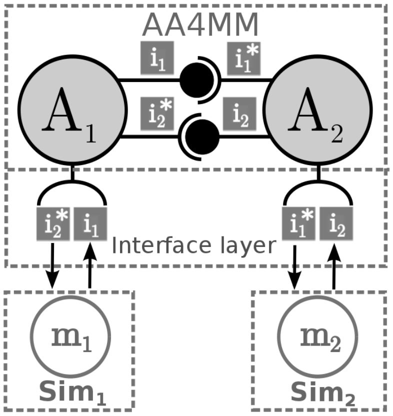
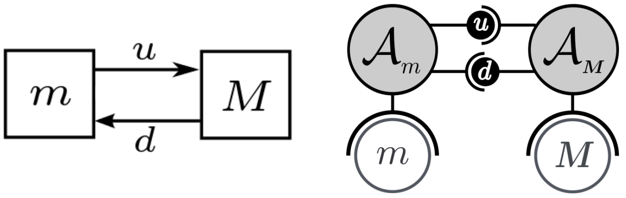


Figure 9. The complete implementation of a multi-model with AA4MM. Each little grey square represents a data type to be defined.

3.4 Multi-level within AA4MM

The framework for multi-level representation presented in section 2.3 is instantiated as follows in the AA4MM's meta-model. As described above, several steps are needed to create a multi-model with AA4MM.

The first step is to define the interaction graph between models where the exchanged data are specified. In our case, each level is associated with a model (*m* for micro and *M* for macro). The interactions are the upward relation from micro to macro (*u*) and the downward relation from macro to micro (*d*) (see Figure 10a).



(a) (b)

Figure 10. Multi-level within AA4MM. (a) Graph of relations between models. (b) The corresponding AA4MM diagram of multi-level representation.

The next step is to describe this graph with the concepts of AA4MM. Each model is associated with a m-agent and an interface artifact. The A_m m-agent manages the micro model and the A_M m-agent manages the macro model. Each relation between models corresponds to a coupling artifact between the m-agents (Figure 10b). We define upward and downward artifacts in order to represent the same named functions. The upward artifact (resp. downward) receives data from the A_m (resp. A_M) m-agent; these data are modified by the upward (resp. downward) function and send to A_M (resp. A_m).

We extended the original AA4MM's specifications in order to develop the corresponding middleware for multi-level modeling. Therefore, we added constraints specific to multi-level representation in respect of the original AA4MM's hypothesis allowing enabling the deployment and implementation of a multi-level model.

The upward and downward functions of our multi-level framework (Section 2.3) are respectively implemented by the transformation functions of the upward and downward artifact with the following new specifications. In a structural coupling between two models, i and i* are lists of values of the same length, therefore length(i) = length(i*). In a upward relations, there is reduction of information, and therefore length(i_m) > length(i_m*). For example, i corresponds to pedestrians and i* corresponds to groups of pedestrians. Similarly, the downward relation d is such that length(i_m) < length(i_m*).

Moreover, to represent the appearance and disappearance of macro entities in a multi-level modeling, we extend the m-agent specifications in order to allow creation and suppression of entities in addition of the updating operation.

An advantage of this framework's instantiation is that, as each conceptual elements of the framework is instantiated by one AA4MM's component, we can use the modularity of the framework in the AA4MM's meta-model in order to rapidly implement the different configurations of Fig.3. This point is developed in the following section.

4. PROOF OF CONCEPTS

In this section, we illustrate through a collective motion example how to implement different patterns of interactions between levels.

The micro level corresponds to moving entities. The micro dynamics corresponds to the flocking model of Netlogo (Wilensky 1998). A micro entity is defined by an identifier, a position, an orientation and a color. Its behavior (Reynolds 1987) is determined by three rules: cohere, align and separate toward other micro entities. The micro entities move at constant speed and have initially the same color that remains unchanged during movements.

The macro level corresponds to moving groups. A group is defined as a set of "close enough" micro entities with similar orientations. In the macro model, a group has an identifier, a position, an orientation, a size and a color (different for each group). We arbitrarily define the group behavior by three rules: cohere, align and separate toward other macro entities (the same as the ones from micro level).

In the experiments, the *f* and *g* functions are each implemented by a Netlogo instance simulating the dynamics. The advantage of the AA4MM meta-model here is that, as an interface artifact is specific to a simulator, we only had to define once the interface artifact for all the following cases.

For sake of clarity, we describe multi-level modeling where spatial scale is the same at the micro and macro levels. Details for the management of heterogeneous spatial and temporal scales with AA4MM can be found in (Siebert 2010b).

In the next parts, we implement different patterns of interactions between these two levels. Each pattern is expressed in a generic way as a graph using the functions *u, d, f, g.* Based on this graph, the corresponding AA4MM diagram is systematically described. Then, the software implementation of this graph is performed thanks to the AA4MM middleware in which we add the domain specific functions.

In section 4.1, we describe how the micro level dynamics influences the macro level: groups form, move and separate according to micro entities configurations (corresponding to the framework's configuration of Fig. 3a).
In section 4.2, we add an influence from the macro level to the micro one: groups determine the color of their composing entities (corresponding to the framework's configuration of Fig. 3b).
In section 4.3, we introduce mutual influences between the macro and micro levels: micro entities form groups (as previously) but groups have their own behavior and impose their movement to their composing entities (corresponding to the framework's configuration of Fig. 3d).
In section 4.4, contrarily to three previous examples, the models don't have the same time scale: one time step at the macro level equals four time steps at the micro one.
In section 5, we discuss what are the advantages of this modular approach for the study of multi-level phenomena.

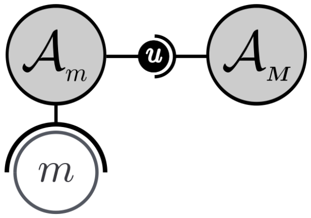
4.1 Micro dynamics and upward relationship

The first example is a selective viewing case (corresponding to the framework's configuration of Fig. 3a), where we want to study the formation/separation of groups over time.

Micro entities (*X*) move (*f*) and form groups. *X* and *f* correspond to a Netlogo instance. An interpretation function u detects groups. It transforms micro entities (*X*) into groups (*Y*). Since groups haven't their own behaviors at the macro level, we only need to represent the different states of the macro level but not their evolution: we don't need a simulator instance. Figure 11a shows the corresponding diagram of relations between the levels in our framework.

$$\begin{tikzcd}
Y_t & Y_{t+dt} \\
X_t \arrow[u, "\boldsymbol{u}"] \arrow[r, "\boldsymbol{f}"] & X_{t+dt} \arrow[u, "\boldsymbol{u}"]
\end{tikzcd}$$

(a)



(b)

Figure 11. A simple upward relation case. (a) Graph of relation between levels of representation and temporal evolution (b) The corresponding multi-level coupling in AA4MM.

Each level corresponds to a m-agent (Figure 11b). A_m manages the micro model *m* in order to simulate the micro dynamics thanks to a Netlogo instance.

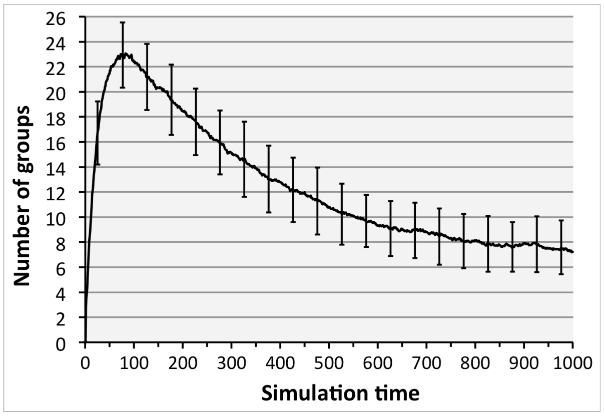


Figure 12. An example of use: studying the number of groups detected in a flocking phenomenon according to simulation time (average over hundred simulations). Error bars indicate standard deviation.

The upward function is implemented as a cluster identification algorithm (Ankerst 1999). It transforms, at each time step, a list of individuals' position and orientation into groups. This function takes two parameters: proximity and orientation thresholds.

At macro level, A_M collects the number of groups detected by the upward artifact at each time step. This way, we can study the evolution of the number of groups as a function of time (Figure 12).

4.2 Micro dynamics, upward and downward relationships

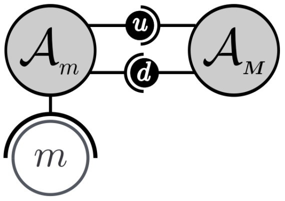
In order to visually identify what are the entities that compose the detected groups, we assign different colors to the micro entities according to the group they belong to. The macro entities have now an influence on the micro level.

Similarly to section 4, micro entities (*X*) move (*f*) and form groups. These groups are detected (*Y*) with an upward function (*u*). As the macro level has an influence on the micro level, a downward function (*d*) is required. The corresponding diagram of relations between the levels of representation and the temporal evolution of the phenomenon can be seen in Figure 13a.

A_M receives data on detected groups from the upward artifact. It assigns each detected group to a color and provides a list of groups and colors to the downward artifact (Figure 13b).

$$\begin{tikzcd} Y_t \arrow[dr, "\boldsymbol{d}"] & Y_{t+dt} \\ X_t \arrow[u, "\boldsymbol{u}"'] \arrow[r, "\boldsymbol{f}"] & X_{t+dt} \arrow[u, "\boldsymbol{u}"'] \end{tikzcd}$$

(a)



(b)

Figure 13. A case of micro dynamics with upward and downward relations. (a) Graph of relation between levels of representation and temporal evolution (b) The corresponding multi-level coupling.

The downward function is implemented in the downward artifact. It assigns groups' color to their composing micro entities and makes this information available to A_m, which changes the color of the micro entities accordingly (Figure 14).

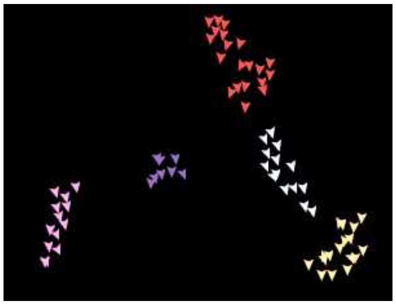


Figure 14. An example of use: visualization of groups on a snapshot of micro level with colors.

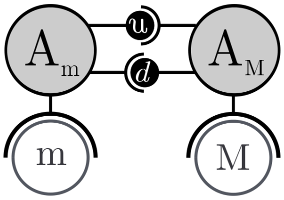
4.3 Micro and macro dynamics, with upward and downward relationships.

We want to model simultaneously the macro and micro dynamics. The micro entities and the groups have their own behaviors, but when a micro entity belongs to a group, the macro dynamics determines its movement: a micro entity moves with the same vector and has the same orientation as its group. If a micro entity is not in a group, its behavior is the same as in section 4.1 and 4.2.

As previously, micro entities (*X*) move (*f*) and form groups. These groups are detected with an upward function (*u*). The detected groups are reified (*Y*) and move (*g*). Y and g are implemented by another NetLogo simulator. *d* is an interpretation function of groups' mouvements. It transforms *Y* (movements of groups) to *X* (movements of entities belonging to groups). The corresponding diagram of relations between the levels of representation and the temporal evolution of the phenomenon can be seen in Figure 15a. Figure 15b is the corresponding representation in the AA4MM meta-model.

$$\begin{tikzcd}
Y_t \arrow[r, "\boldsymbol{g}"] \arrow[dr, "\boldsymbol{d}"] & Y_{t+dt} \\
X_t \arrow[u, "\boldsymbol{u}"] \arrow[r, "\boldsymbol{f}"] & X_{t+dt} \arrow[u, "\boldsymbol{u}"]
\end{tikzcd}$$

(a)



(b)

Figure 15. A case of micro dynamics, macro dynamics, upward and downward relations. (a) Graph of relations between levels of representation and temporal evolution (b) Corresponding multi-level coupling.

A_m manages the micro model *m* and simulates its dynamics.

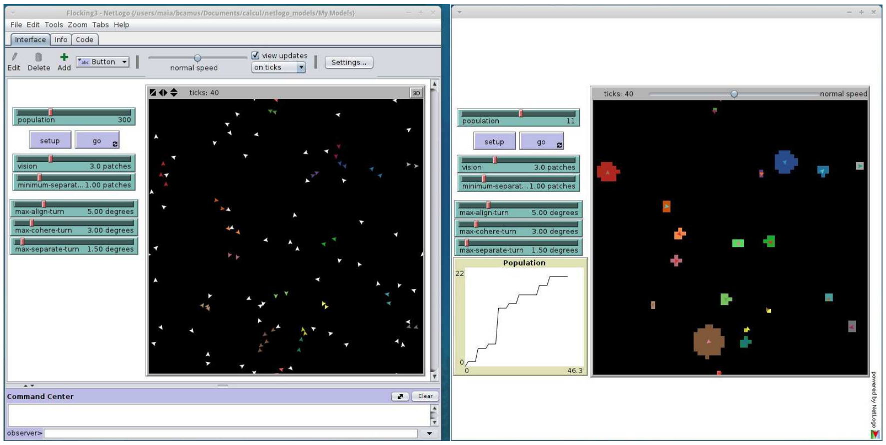
The upward function of section 4.1 and 4.2 is extended. It sets detected groups' attributes by aggregating micro data as follow: position is the gravity center of the micro cluster, orientation corresponds to the average direction of its constituent micro entities and size refers to the dispersion.

A_M manages the macro model *M* and its dynamics. It receives detected groups from the upward artifact and adds, removes or updates them in *M,* executes one step of simulation, collects data and sends group's vectors and orientation to the downward artifact.

The downward function transforms groups' movements into entities' movements. It is implemented in the transformation function of the downward artifact.

A_m receives from this artifact the movements of some entities (those that belong to a group). It modifies the states of these entities accordingly. For the entities not belonging to a group, it asks for a simulation step in order to update their states.

Figure 16 is a snapshot of the multi-model execution. Figure 17 shows an example of use of this pattern of interactions.



population 300
vision 3.0 patches
minimum-separat... 1.00 patches
max-align-turn 5.00 degrees
max-cohere-turn 3.00 degrees
max-separate-turn 1.50 degrees

population 11
vision 3.0 patches
minimum-separat... 1.00 patches
max-align-turn 5.00 degrees
max-cohere-turn 3.00 degrees
max-separate-turn 1.50 degrees

Figure 16. Snapshot of the multi-model execution in a micro and macro dynamics, with upward and downward relationships case. (left: micro level, right: macro level)

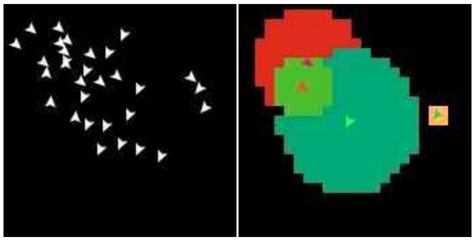


Figure 17. An example of use : study the groups' crossing at micro (left) and macro (right) levels.

4.4 Different time scales.

In this section we describe how AA4MM's framework manages different temporal scales between the two simulators. In this case, one macro simulator's time step corresponds to four micro simulator's time steps. This difference of temporal scale implies that, for the same simulation time period, while the macro model produces one data, the micro model produces four ones. Thanks to the modularity of AA4MM, only the downward artifact has to be changed when taking the previous configurations back: the downward function linearly decomposes now groups' movement vectors it received in four parts. The AA4MM's proved decentralized algorithm automatically managed the models synchronization in this case: only the last of the four micro data received by the upward artifact is considered by the macro model.

5 Discussions

We can see with the different examples of use (Figure 12, 14 and 17) that our approach preserves the advantages of a multi-level representation. We can study the formation of higher-level structures (Figure 12), enhance the visualization of the phenomenon (Figure 14), and make explicit new phenomenon, for instance group's crossing (Figure 17).

This modular vision of multi-level modeling facilitates the study of the multi-level phenomenon. As levels' relationships are reified in an AA4MM multi-level coupling, we can test hypothesis about the level's influences by comparing different multi-level representations of the same phenomenon. It is then possible to study the mutual influence of levels, for instance the mutual influence of individuals and groups. As shown in section 4, we can rapidly switch from a pattern of interaction to another with well-focused efforts. Figure 18 shows a comparison of the number of groups detected in simulations using different level's dynamics (described by Table 2) and influences.

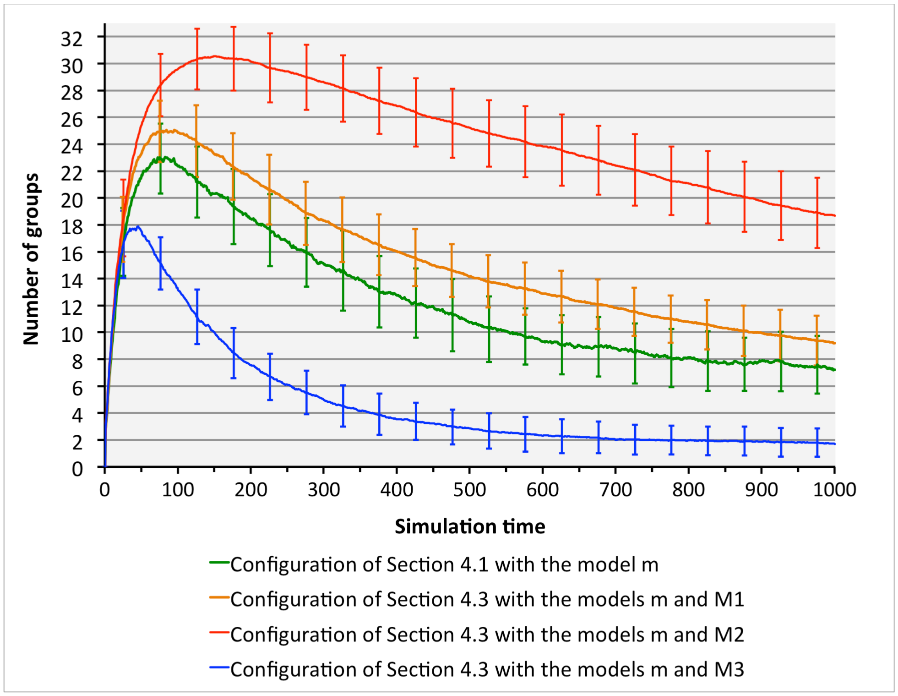


Figure 18. Number of groups detected according to simulation time (average over hundred simulations) in different AA4MM configuration. Error bars indicate standard deviation.

Models	**Cohere parameter**	**Align parameter**	**Separate parameter**
m	3	5	1.50
M_1	3	5	1.50
M_2	0	0	10
M_3	14,25	14,25	0

Table 2. Values of the parameters of the different models being compared. m correspond to the micro model and M_i to models of the macro dynamics.

The AA4MM's advantages are also preserved with a multi-level coupling. Thanks the modular approach and a clear separation of the different levels of questions, we can change the simulator implementing one model with only changing the interface artifact. It is then possible to compare different implementations of the same model. AA4MM can also be the support to an incremental modeling process as it is possible to test different versions of the same model and check their interoperability in the multi-model.

6. CONCLUSION

In this article, we proposed to consider a multi-level model as a multi-model. This approach relies on a multiagent vision of a multi-model. We used the AA4MM meta-model to design and instantiate the multi-model.

We defined a framework that organizes the multi-level's concepts and their relations. This framework specifies the patterns of interaction between levels. We translated this framework into the AA4MM while respecting the semantic constraints specific to the simultaneous representation of two levels.

As a proof of concept, we used a collective motion phenomenon to detail different kinds of multi-level representation.

Since we extended AA4MM, we can benefit of its advantages for multi-level representations (reuse of simulators, of meta model component, ease to switch from one pattern to another, ...).

We limited to two simultaneaous levels in this article. In future works, two directions can be addressed: representing several levels simultaneouly or representing a phenomenon with several levels but considering only one at a time.

REFERENCES

ANKERST M., Breunig M., Kriegel H-P., Sander J. (1999). "OPTICS: Ordering Points to Identify the Clustering Structure". ACM SIGMOD International Conference on Management of Data. ACM Press. pp. 49–60.

BOURGINE P. (2008). “Distributed Problem Solving in Natural and Artificial Complex Systems.” Invited talk in Engineering Societies in the Agents World IX, 9th International Workshop, ESAW.

CETINKAYA D., Verbraeck A., Seck M. D. (2010). A metamodel and a DEVS implementation for component based hierarchical simulation modeling.01/2010. In proceeding of: Proceedings of the 2010 Spring Simulation Multiconference, SpringSim 2010, Orlando, Florida, USA, April 11-15.

DAVIS, P. K. and Hillestad, R. J. (1993). Families of models that cross levels of resolution: Issues for design, calibration and management. In Proceedings of the 1993 Winter Simulation Conference, ACM, New York.

DEMANGE, J. (2012) .Un modèle d’environnement pour la simulation multiniveau : application à la simulation de foules. Phd Thesis, Belfort-Montbéliard University (in French).

DUBOZ R., Ramat E., Preux P. (2003). Scale transfer modelling : using emergent computation for coupling an ordinary differential equation system with a reactive agent model. System Analysis, Modelling, Simulation, 43(6), 793-814.

EL HMAM M., Abouaissa H., Jolly D., Benasser A. (2006). “Macro-Micro Simulation of Traffic Flow”. 12th IFAC Symposium on Information Control Problems in Manufacturing, INCOM, Saint-Etienne, France.

FERBER J., Gutknecht O. (1998). Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems, ICMAS 98 (International Conference on Multi-Agent Systems), Paris, Y. Demazeau (ed), IEEE Press, pp. 128-135.

GAUD N., Galland S., Gechter F., Hilaire V., and Koukam A. (2008). “Holonic Multilevel Simulation of Complex Systems: Application to Real-time Pedestrians Simulation in Virtual Urban Environment.” Simulation Modelling Practice and Theory, 16:1659–1676.

GIL-QUIJANO, J., Louail T., Hutzler G. (2010). “From Biological to Urban Cells: Lessons from Three Multilevel Agent-Based Models.” PRIMA 2010: 620-635.

KLÍR, J., Šafarik, J. (2004). Variable resolution modeling in interactive parallel discrete event simulation. In *Electronic computers and informatics.* Košice: VIENALA Press, s. 353-358. ISBN: 80-8073-150-0

LECLERC T., Siebert J., Chevrier V., Ciarletta L., Festor O. (2010). “Multi-modeling and Co-simulation-based Mobile Ubiquitous Protocols and Services Development and Assessment.” In : 7th International ICST Conference on Mobile and Ubiquitous Systems - Mobiquitous 2010, Australie Sydney.

MUSSE S.R., Thalmann D. (1997). “A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis,” Proc. Workshop Computer Animation and Simulation of Eurographics. '97.

NATRAJAN, A., Nguyen-Tuong, A. (1995). “To disaggregate or not to disaggregate, that is not the question” , ELECSIM, Internet, April-June, 1995, Technical Report CS-95-18, Department of Computer Science, University of Virginia.

REYNOLDS C. (1987). "Flocks, Herds and Schools: A Distributed Behavioral Model.". SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques (Association for Computing Machinery): 25–34, doi:10.1145/37401.37406, ISBN 0-89791-227-6.

REYNOLDS, F. P., Natrajan, A., Srinivasan S. (1997). “Consistency maintenance in multiresolution simulation.” ACM Transactions on Modeling and Simulation, 7(3) pp. s368-392.

RICCI A., Viroli M., Omicini A. (2007). “Give Agents their Artifacts: the A&A Approach for Engineering Working Environments in MAS.” In AAMAS '07: Proc. of the 6th international joint conference on Autonomous agents and multiagent systems, pp601-603, New York, USA.

RODRIGUEZ S., Gaud N., Hilaire V., Galland S., Koukam A. (2007). “An analysis and design concept for self-organization in holonic multi-agent systems. In ESOA’06, Workshop of AAMAS.

ROPELLA G.E., Hunt C.A., Nag D.A. (2005). "Using Heuristic Models to Bridge the Gap Between Analytic and Experimental Models in Biology." Proceedings of the 2005 Agent-Directed Simulation Symposium (San Diego, CA, April 2-8), Simulation Series vol. Vol. 37.

SECK, M. D., H. Job Honig. (2012). "Multi-perspective Modelling of Complex Phenomena." Computational & Mathematical Organization Theory: 1-17.

SIEBERT J., Ciarletta L., Chevrier V. (2010a). “Agents & Artefacts for Multiple Models Coordination: Objective and Decentralized Coordination of Simulators.” In Proceedings of the 2010 ACM Symposium on Applied Computing. pp2024-2028.

SIEBERT J., Ciarletta L., Chevrier V. (2010b). “Agents and Artifacts for Multiple Models Co-evolution. Building Complex System Simulation as a set of Interacting Models.” In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, in proc. of AAMAS 2010, May 10, 2010, Toronto, Canada, pp509-516.

SIEBERT J. (2011). “Approche Multi-agent pour la Multi-modélisation et le Couplage de Simulations. Application à l'étude des Influences entre le Fonctionnement des Réseaux Ambiants et le Comportement de leurs Utilisateurs.” Phd Thesis, Henri Poincaré University - Nancy 1, (in French).

SPRINKLE J., Rumpe B., Vangheluwe H., Karsai G. (2010). Metamodelling State of the Art and Research Challenges. In Proceedings of the 2007 International Dagstuhl conference on Model-based engineering of embedded real-time systems. Pages 57-76 . Springer-Verlag Berlin, Heidelberg.

VO D., Drogoul A., Zucker J-D. (2012). “An Operational Meta-model for Handling Multiple Scales in Agent-based Simulations.” Research, Innovation, and Vision for the Future (RIVF), Ho Chi Minh: Viet Nam.

WILENSKY U. (1998). “NetLogo Flocking Model”. http://ccl.northwestern.edu/netlogo/models/Flocking. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

YILMAZ L. and Ören, T.I. (2004). Dynamic model updating in simulation with multimodels: A taxonomy and a generic agent-based architecture. In Proceedings of SCSC 2004 - Summer Computer Simulation Conference, July 25-29, 2004, San Jose, CA., pp. 3-8.

YILMAZ L., Lim A., Bowen S., Oren T. (2007). "Requirements and Design Principles for Multisimulation with Multiresolution, Multistage Multimodels." Simulation Conference, 2007 Winter. IEEE.