A Flexible Approach to Multi-Level Agent-Based Simulation with the Mesoscopic Representation

Laurent Navarro
Thales / UPMC – LIP6[1]
Paris, France
laurent.navarro@thalesgroup.com

Vincent Corruble
UPMC – LIP6[1]
Paris, France
vincent.corruble@lip6.fr

Fabien Flacher
Thales
Vélizy, France
fabien.flacher@thalesgroup.com

Jean-Daniel Zucker
IRD – UMMISCO[2]
Bondy, France
jean-daniel.zucker@ird.fr

ABSTRACT

Large-scale simulations often use multiple agent representations to permit the study of specific multi-agent phenomena, and to find a balance between run-time performance and level of detail of the simulation. Although these approaches are effective, they do not always offer the desired level of analysis, especially when this level is between the resolutions of the models available. In this paper, we aim at offering a finer method in exploring this trade-off by introducing an intermediate level between two given resolutions, which can apply to all agent models and allows a more progressive transition to offer the desired level of analysis. We introduce a framework for such a methodology and evaluate it through the extension of an existing approach, along two criteria: its impact on computational resources, and an estimate of the dissimilarity between a simulation using our methodology and one without. Initial experiments show that consistency is almost maintained while CPU gain varies from low to significant depending on the context.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence – Multiagent systems

General Terms

Algorithms, Performance, Experimentation.

Keywords

Agent-based simulations – Simulation techniques – Tools and environments – Level of Detail – Multi Representation Modeling.

1. INTRODUCTION

The simulation of human behavior has attracted growing interest in many applicative fields. Its ability to populate complex environments with agents showing credible behaviors allows the study of different scenarios in fields such as security, infrastructure protection, urban planning or video game. The realism of the interactions exhibited by the virtual agents is a key point for military and civilian training simulations. However, those applicative domains also require the animation of large numbers of agents to breathe life into their scenarios. This ability, coupled with the complexity of the underlying agent models, requires important computational power. Indeed, microscopic behavioral models such as those used for computing the emotional, motivational, navigational and cognitive states of a single agent need several updates per second and sometime additional data from surrounding agents, for detection and communication purposes. Thus, the computational load is directly linked to the number of simulated agents.

Several multi-resolution approaches have tried to overcome this difficulty by introducing new representation levels for the simulated entities, besides the microscopic one. From purely macroscopic to hybrid representations, they usually aim at breaking the typical one-to-one match between simulated entities from the "real world" and virtual agents in the simulation, so as to optimize the use of computing resources. However, their implementation is often done at the expense of a significant design cost. Moreover, the lower-resolution models do not always allow the reproduction of microscopic phenomena, thus impacting significantly the simulation consistency. This limitation has been highlighted in the work described in [7], which is the basis of the approach presented here. In [7], a dynamic agent aggregation method applied to a pedestrian simulation showed that, in specific contexts, the aggregates were not able to exhibit specific navigational and decisional patterns, which were yet observed in the microscopic simulation of the same scenario.

A possible and natural way to overcome this is to enhance the lower-resolution models, allowing them to reproduce the desired phenomena. However, this solution requires the ability to understand and modify the models as well as the associated transition functions. In addition, it is not reasonable to apply such process each time a new phenomenon emerges from the microscopic level without complicating the lower-resolution models. On the contrary, we argue that it is possible to offer a more flexible and controlled way to migrate from a representation level to another in order to keep, if needed, parts of the highest-resolution model responsible for the emergence of the desired phenomena, while abstracting less needed behavioral aspects.

To do so, we present here a novel approach for multi-level agent-based simulations, by introducing an intermediate level between microscopic and macroscopic resolutions, which can apply to all agent models. This level, referred here as *mesoscopic*, allows a more progressive transition between two models to offer the desired level of analysis given the context of the simulation and the user needs. With this aim, we define the notion of agent processes, which are inner parts of the agent models. Then, we extend the generic notions of dynamic change of representation and spatial aggregation introduced in [7]. Finally, we define

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.
Copyright © 2013, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

[1] Université Pierre et Marie Curie – LIP6, 4 Place Jussieu, 75005 Paris FR

[2] Institut de Recherche pour le Développement, UMI 209, UMMISCO, IRD France Nord, F-93143, Bondy, FR

several environments in which we evaluate the approach experimentally, discuss the results and propose enhancements for future work.

2. RELATED WORK

Finding the resolution that best suits a given problem, among several models of a given phenomenon, has been widely studied within the Multi Representation Modeling (MRM) field through the joint execution of multiple models. In selective viewing, only the highest resolution model is executed at all times, and all other models are emulated by abstracting the representation of the most detailed one [9]. This approach is used when the simulation requires a phenomenon to be modeled in detail. Although it may be efficient for applications which need high precision, it requires huge computational resources. Moreover, multiple models are not necessarily organized hierarchically in a natural way, preventing designers to define which model is the most detailed. Finally, executing the higher-resolution model does not always facilitate decision making.

In aggregation / disaggregation techniques, only one model is executed at a given time, but not necessarily the most detailed one. Aggregation corresponds to the transition from high-resolution entities to a low-resolution one, while disaggregation is the opposite process. The choice of model depends on the user's need or the necessity to match the resolution of other interacting entities. Several variants exist, such as Full Disaggregation [10], Partial Disaggregation [6], Playboxes [11], Dynamic Component Substitution [15] and Pseudo Disaggregation [10]. Each of them can lead to speedup when a balance between complexity and simulation needs is found. But they require huge resources when moving from one model to another, and problems – such as chain disaggregation – may arise in case of cross-level interactions. Finally, transition latency, network flooding and thrashing may impact simulation consistency [16].

Variable Resolution Modeling (VRM) allows the creation of families of models that support dynamic changes in resolution [9] by introducing several constraints. Thus, all the models parameters are standardized within a dictionary and inserted in a hierarchical structure symbolizing their dependencies. Rules are defined between models to match the computation time steps, ensure the consistency of the simulation and allow the calibration. Following those rules, a designer can create a family of models that can adapt their resolution level to the simulation needs. But this approach is mainly theoretical and is not suitable when the models are pre-designed and cannot be adapted to the VRM approach.

Multiple Representation Entities [5] is a last example from the MRM field that is of particular interest here. This approach maintains, at all time in the simulation, all representations through all available models of a given phenomenon, using appropriate mapping functions to translate changes between two representations. This allows interactions between all the representations, and avoids wasting resources as happens when scaling from one model to another. MRE is a powerful way to deal with complex MRM, which offers a remedy for the weakness of aggregation / disaggregation methods and requires lower resources than the simultaneous execution of multiple models. But it only provides mathematical requirements for the mapping functions, through the use of attributes dependency graphs. Also, it does not identify the representation at any level nor relationships between representations.

Some approaches in Multi-Agent Simulation (MAS) also leverage the principle of simultaneous use of microscopic and macroscopic models, by partitioning the environment and running a different model in each zone. The pedestrian simulations described in [1] and [17] use high-level flow and distribution models to steer non-visible agents along a network of nodes that describe the accessible areas of a city, and a microscopic collision avoidance model with speed adjustment for visible actors. Similarly, [13] and [14] describe traffic simulations using a static predesigned world. Thus, a macroscopic model based on the flow theory is used in low interest areas without crossroads, and a microscopic multi-agent car-following model in high interest areas. Those architectures can handle several thousand agents with high consistency and offer a good interactivity with the agents' behavior within both macroscopic and microscopic areas. But they require a preprocessed environment and predefined transition functions between the agent models.

A last set of MAS approaches tries to overcome the design constraint of the transition functions between agent models. In IVE [18], the 2D environment is abstracted in a topological tree in which each leaf represents a reachable location and each node an aggregation of areas. On the other hand, the agents are driven by a hierarchical reactive planning mechanism whose nodes are linked to the ones of the topological tree. IVE can then adapt the level of detail of the agent's behavior according to the importance of the area in which the entity is located without having to tweak the model of change the agent's representation. Similarly in JEDI [4], the environment is represented as atomic cells nested together to form lower-level areas. To each cell is linked a matrix of possible interactions, thus defining the behaviors of the agents it contains. The interactions of the low-resolution areas are deducted from the ones of the cell it aggregates. These two frameworks can change the representation level of their agents without using transition functions, and thus maintain a strong consistency. However, this is only possible because the behavioral models are rule-based, and thus remain quite simple.

All the approaches described above use a set number of representations that depends on the number of models they consider. They therefore cannot address accurately a problem whose solution is intrinsically situated between two representation levels. For a given resolution, the behavior of a virtual actor depends entirely on its assigned agent model. If its dynamics diverges from the one of the other higher or lower resolution models, then the consistency of the simulation cannot be maintained, regardless of the transfer functions used.

3. DYNAMIC LEVEL OF DETAIL

This section describes more precisely an alternative approach to the ones reviewed above, and provides a formal and experimental context to the rest of this paper. This approach, detailed in [7], introduces an aggregation method for the dynamic (online) migration from several microscopic agents to a single macroscopic entity. It is composed of two phases: an *agent aggregation* phase and a *representation change* phase.

The first phase is used to determine which agents should be grouped together, based on the context of the simulation. Given two entities A_1 and A_2, this approach introduces a physical distance, $D_\varphi(A_1; A_2)$, and a psychological one, $D_\psi(A_1; A_2)$,

which are combined to estimate the affinity $Aff(A_1; A_2)$ between those two agents. The affinity represents the proximity of A_1 and A_2, based on their physical and mental states. The context of the simulation is taken into account with the definition of events, which are points of interest requiring surrounding agents to be simulated at highest resolution to ensure simulation consistency. If $E = \{E_1; \dots; E_m\}$ is a set of events, the affinity of A_1 and A_2 regarding the simulation context is denoted by $Aff(A_1; A_2; E)$, such as:

$$Aff(A_1; A_2) = f\left[D_\varphi(A_1; A_2); D_\psi(A_1; A_2)\right]$$

$$Aff(A_1; A_2; E) = \underset{i \in [\![1; |E|]\!]}{\text{Max}} \left[g\left[D_\varphi(A_1;\ A_2;\ E_i); D_\psi(A_1;\ A_2;\ E_i)\right]\right]$$

$$with \begin{cases} D_\varphi(A_1; A_2;\ E_i) = \text{Min}\left[D_\varphi(A_1; E_i);\ D_\varphi(A_2; E_i)\right] \\ D_\psi(A_1; A_2; E_i) = \text{Min}\left[D_\psi(A_1; E_i);\ D_\psi(A_2; E_i)\right] \end{cases}$$

$Aff(A_1; A_2)$ and $Aff(A_1; A_2; E)$ are combined to define the aggregation utility $U_{Ag}(A_1; A_2)$. It represents the usefulness of creating an aggregate considering A_1, A_2 and the simulation context. A similar process is described to disaggregate a low resolution entity A', based on physical and psychological distances between A' and E, which are combined to set a disaggregation utility $U_{Disag}(A')$.

In a second phase, the representation change provides a way to compute the attributes of the representation of a low resolution agent given those of several higher resolution entities, and vice versa. Let M_1 be an agent model, which is a computational abstraction of the global behavior of a synthetic actor. M_1 takes as input the representation of an agent and of its environment, and outputs an action or a modification of the agent's representation. This representation – denoted by $Rep(M_1)$ – is the set of attributes $\{\mathcal{A}_i\}_{i \in [\![1..|M_1|]\!]}$ needed by the agent model to perform its task and is usually assimilated to its internal state. Then, the representations of an agent A_1 in M_1 at time t, and a set of agents $A = \{A_1;\ \dots;\ A_n\}$ in M_1 at time t are the vector and the matrix of attributes' values, denoted by $Rep(A_1; M_1; t)$ and $Rep(A;\ M_1; t)$, such as:

$$Rep(A_1; M_1; t) = \begin{pmatrix} a_{1;\,1}(t) \\ \vdots \\ a_{1;\,|M_1|}(t) \end{pmatrix}$$

$$Rep(A;\ M_1; t) = \left(Rep(A_1;\ M_1; t); \dots;\ Rep(A_n;\ M_1; t)\right) = \begin{pmatrix} a_{1;\,1}(t) & \cdots & a_{n;\,1}(t) \\ \vdots & \ddots & \vdots \\ a_{1;\,|M_1|}(t) & \cdots & a_{n;\,|M_1|}(t) \end{pmatrix}$$

If M_2 is another agent model whose resolution is lower than the one of M_1, then the objective of the representation change axis is to compute the representation of the aggregate A' in M_2 at time t from the set of agents A, such as:

$$Rep(A';\ M_2; t) = F_{Ag}[Rep(A;\ M_1; t)]$$

To do so, the approach proposes to divide the representation of each model into subsets of attributes sharing the same meaning and therefore a common dynamics. Accordingly, F_{Ag} is split into several sub functions, each operating on a specific class of attributes. In [7], those sub functions are simple **SUM**, **MIN**, **MAX**, **MEDIAN** or **MEAN** operators. During aggregation, a memory function is used. Its goal is to save data for further disaggregation, for each attribute of each agent in A and regarding the computed representation of A', such as:

$$Mem(A;\ M_1; t) = F_{Mem}[Rep(A;\ M_1; t);\ Rep(A';\ M_2; t)]$$

$$Rep(A;\ M_1; t') = F_{Disag}[Rep(A';\ M_2; t');\ Mem(A;\ M_1; t)]$$

The approach described above has been evaluated on a large-scale multi-agent simulation, along two criteria: the gain in computer resources induced by the dynamic level of detail, and the preservation of the simulation consistency. This second criterion is described in [9] and represents the quantity of information lost during the aggregation / disaggregation process. It is obtained by comparing the final states of a microscopic simulation and a multi-resolution one. Evaluations conducted on different environments have shown that the approach offers, in most scenarios, an effective way to save computational resources at the expense of a small loss of consistency. However, several experimentations highlighted its failure at accounting for macroscopic phenomena emerging at the microscopic level. For example, specific navigational and decisional effects in crowded environments due to congestion, such as the increase in physical collisions and stress resulting in panic and fleeing behaviors, are not reproduced.

This is mainly a consequence of the fact that the same agent model is used at both microscopic and macroscopic representation levels. Thus, the physical space occupied by an aggregate is the same as the one of a microscopic agent. In the multi-resolution simulation, and for the same number of agents as in the microscopic one, the congestion of the environment no longer exists. Entities' perceptions are different from one simulation to another, which leads to diverging choices of paths and actions. The obvious inability of the agent model to reproduce a spatial phenomenon at low-resolution is sufficient to induce a significant and unacceptable loss of consistency. While a possible solution is to create an adequate macroscopic model capable of managing accurately the space taken by an aggregate, we suggest here another approach which is to preserve the spatiality of the aggregated agents, thus solving the problem described above, while migrating their decision-making processes at the macroscopic level. We chose to explore and generalize this idea in this paper, as described below.

4. MESOSCOPIC LEVEL

4.1 The mesoscopic representation

In this work, we aim to offer a smoother transition between two representations by creating an intermediate resolution level. Unlike the macroscopic one in which all agents are aggregated into a single entity, the mesoscopic level centralizes parts of the computation performed on the microscopic agents in order to free computational resources while letting other parts be updated according to their initial level.

We consider the case where the higher-resolution model is not composed of a single block, such as a single function, but rather a set of several distinct processes, each being a mostly autonomous module leading to a particular skill of the agent. Such skills can be the navigation, decision, emotions, planning, communication or social interactions ability of the agent. For example, in well-known cognitive architectures such as ACT-R [3], ICARUS [2] and SOAR [8] the processes could be the emotional, decisional or sensitive / short term / long term memories modules for the first one, or the declarative procedural memory, pattern matching, and production execution modules for the second. Those processes are themselves models, taking as inputs a subset of the agent model's inputs and outputting modifications of the agent's representation

as well as specific data. They are usually chained, each of them requiring others to do their work before it can execute its own. To continue the previous example, the SOAR decision process needs the elaboration process to fire all production rules, meaning that the working memory has been previously updated by the perception module. Thus, it is often possible to identify a hierarchy of dependencies between processes within a single agent model.

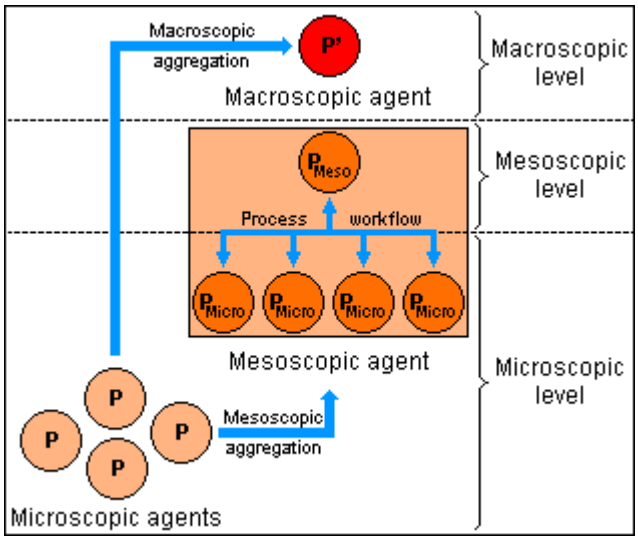

Figure 1. Example of the macroscopic and mesoscopic aggregation of 4 agents implemented as a set of P processes.

We consider that the agent model M is composed of a set of processes $P = \{P_1;\ P_2;\ \dots;\ P_m\}$. The goal, outlined in Figure 1, is to allow the subset $P_{meso} \subset P$ to be run at the mesoscopic level while $P_{micro} = P \setminus P_{meso}$ will remain at the microscopic level. The microscopic representation of an agent A_1 and the one of a set of agents $A = \{A_1;\ A_2;\ \dots;\ A_n\}$ in $P_k \in P$ at time t are denoted by $Rep_{micro}(A_1; P_k; t)$ and $Rep_{micro}(A; P_k; t)$ such as:

$$Rep_{micro}(A_1;\ P_k; t) = \begin{pmatrix} a^k_{1;1}(t) \\ \vdots \\ a^k_{1;|P_k|}(t) \end{pmatrix} with\ Rep(P_k) \subset Rep(M)$$

$$Rep_{micro}(A;\ P_k; t) = \left(Rep_{micro}(A_1;\ P_k; t); \dots; Rep_{micro}(A_n;\ P_k; t)\right) = \begin{pmatrix} a^k_{1;1}(t) & \cdots & a^k_{n;1}(t) \\ \vdots & \ddots & \vdots \\ a^k_{1;|P_k|}(t) & \cdots & a^k_{n;|P_k|}(t) \end{pmatrix}$$

For a process $P_k \in P_{meso}$, we need to compute the mesoscopic representation of A at time t denoted by $Rep_{meso}(A; P_k; t)$. To do so, we use the methodology described in [7] and detailed above. First, we partition $Rep(M)$ among several attributes classes. Then we link each class with an aggregation operator and its corresponding disaggregation and memory functions, respectively denoted by F_{Ag}, F_{Disag} and F_{Mem}, so that we have for a given process P_k:

$$Rep_{meso}(A; P_k; t) = F_{Ag}[Rep_{micro}(A;\ P_k; t)]$$

$$Mem(A;\ P_k; t) = F_{Mem}[Rep_{meso}(A; P_k; t);\ Rep_{micro}(A;\ P_k; t)]$$

$$Rep_{micro}(A;\ P_k; t') = F_{Disag}[Rep_{meso}(A; P_k; t');\ Mem(A;\ P_k; t)]$$

This method allows the migration from a microscopic representation to a macroscopic one – and vice versa – except that it operates here on a chosen subset of $Rep(M)$. Thus, it allows a single process to work at the mesoscopic level by computing the needed representation of A. However, this process is part of a hierarchy and may have dependencies with other processes. In order to avoid inconsistencies in the computation of the agent model, we must consider the attributes of $Rep(M)$. If an attribute $\mathcal{A}$ is only used at the microscopic level, then it is ignored. On the other hand, if $\mathcal{A}$ is only used at the mesoscopic level, which means that $\mathcal{A} \notin \bigcup_{P_k \in P_{micro}} Rep(P_k)$, it is aggregated once. Finally, if $\mathcal{A}$ is used at both levels, we need to maintain both microscopic and mesoscopic values of $\mathcal{A}$ when it is updated by any process, with the aggregation, disaggregation and memory functions described above for the attributes class to which it belongs. In practice, it is possible to restrict such computation – which can be CPU intensive – by updating the microscopic values of an attribute only if a mesoscopic process has updated it earlier in the agent model update and vice versa.

At this point, we propose a simple example to illustrate the process described above. Let M be an agent model whose representation is composed of 5 attributes and which can be split into a set P containing 3 processes, such as:

$$Rep(M) = \{\mathcal{A}_1;\ \dots;\ \mathcal{A}_5\}$$

$$P = \{P_1;\ P_2;\ P_3\} \text{ and } \begin{cases} Rep(P_1) = \{\mathcal{A}_1;\ \mathcal{A}_2;\ \mathcal{A}_5\} \\ Rep(P_2) = \{\mathcal{A}_2;\ \mathcal{A}_3;\ \mathcal{A}_4\} \\ Rep(P_3) = \{\mathcal{A}_4;\ \mathcal{A}_5\} \end{cases}$$

The dependency hierarchy between the processes constituting our example is outlined in Figure 2. It shows that, in order to achieve a full update of the agent model, P_1 must be computed first, then P_2 and finally P_3. When a mesoscopic agent A is created at time t, $Rep_{meso}(A; P_2; t)$ is computed because P_2 is at the mesoscopic level. At time t', when A is updated in the simulation, then P_1 is computed, followed by P_2. To do so, the value of $\mathcal{A}_2$ in $Rep_{meso}(A; P_2; t')$ is aggregated from $Rep_{micro}(\mathrm{A}; P_1; t')$, because $\mathcal{A}_2$ is common to $Rep(P_1)$ and $Rep(P_2)$. Similarly, in order to update P_3, the value of $\mathcal{A}_4$ in $Rep_{micro}(\mathrm{A}; P_3; t')$ is disaggregated from $Rep_{meso}(A; P_2; t')$. The values of $\mathcal{A}_1$, $\mathcal{A}_3$ and $\mathcal{A}_5$ are not modified by our approach as they only appear in one representation level.

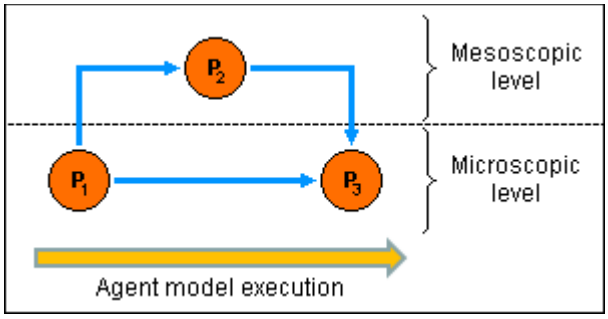

Figure 2. Example of dependency hierarchy of a model constituted of 3 processes.

Such approach allows the migration of any process constituting the agent model from the microscopic to the mesoscopic level, resulting in the freeing of computation time. The choice of the aggregation functions – and their corresponding disaggregation and memory operators – must be done wisely in order to maintain simulation consistency as defined in [9]. The selection of the

processes to transfer is also an important issue. Firstly, it impacts the consistency, as the processes do not share the same impact on the environment and the surrounding agents. Moreover, it is better to migrate those that require high computational resources and have few dependencies with others in order to avoid constant aggregation / disaggregation of attributes. Finally, it is important to note that migrating all processes to the mesoscopic level is equivalent to aggregating the set of agents to a macroscopic one driven by the same agent model.

4.2 Spatial aggregation

This section tackles the problem of finding which agents should be aggregated to form a mesoscopic entity, and which processes of this new entity must be migrated to the mesoscopic level. To do so, our approach uses the mechanisms of spatial aggregation described in Section 3. Thus, it relies on the physical and psychological distances D_φ and D_ψ between the considered pair of agents and the set of simulation events, combined to compute the aggregation utility U_{Ag}. However, the definitions of D_φ and D_ψ are not trivial here because the representation of the mesoscopic agent is fragmented. Indeed, if their computation requires a set of attributes of which a part is used by a mesoscopic process while the rest remains microscopic, then their implementation will require choosing carefully the suitable attributes in the correct representation.

In addition to identifying agents that can be aggregated, this method offers the ability to choose the processes which should migrate to the mesoscopic level and those which should stay at the microscopic one. To do so, we define for each process an aggregation threshold, thus creating a total order over them. It means that the lower the threshold, the higher priority is given to the migration of the process to the mesoscopic level. At aggregation time, those values are compared to U_{Ag} and will determine the structure of the final aggregate. It is important to note that the aggregation thresholds do not rely on the dependency hierarchy described above. However, choosing the processes order according to the hierarchy lowers the risk of having attributes at microscopic and mesoscopic levels which, as seen before, need to be maintained in both representations to ensure the consistency of the processes computation. Of course, it may happen that two processes cannot be separated because of some characteristics of their implementation or of the high number of attributes they share. In this case, a possible solution would be to assign the same threshold to both.

The disaggregation of a mesoscopic agent proceeds of the same idea, via the definition of a disaggregation utility. However, unlike the macroscopic approach where this utility has to be computed once for the whole aggregate, it must here be computed for each microscopic entity composing the mesoscopic agent, because some of its processes might remain at the microscopic level and are involved in the calculus of D_φ and D_ψ. Although this approach requires significant computational resources, it allows disaggregating a single microscopic agent from the aggregate, in regard to the simulation context. Such feature was not possible with the macroscopic approach. For example, if a microscopic agent tries to communicate with some microscopic entities of a mesoscopic agent, and if the communication process is still at the microscopic level, then the disaggregation utility of the communicating entities – and only them – will allow a partial disaggregation of the mesoscopic agent. We then have, for a mesoscopic agent A:

$$Aff(A_i; E) = \underset{i \in [\![1;|E|]\!]}{\text{Max}} [Aff(A_i; E_i)], A_i \in A$$

The method described above only applies when the aggregation and disaggregation utilities between two agents must be computed. However, it does not let the processes migrate dynamically when the mesoscopic agent is alone. To do so, we define a representation change utility for a mesoscopic (or macroscopic) agent A, denoted by $U_{RC}(A)$, such as:

$$U_{RC}(A) = f\left[\underset{i \in [\![1;|E|]\!]}{\text{Max}} \left[f\left[D_\varphi(A;\, E_i); D_\psi(A;\, E_i)\right]\right]\right]$$

U_{RC} has nearly the same meaning as the aggregation utility except that it applies to a single agent. As a result, comparing it to the processes aggregation thresholds lets the aggregate adapt dynamically the representation level of its own processes. While this approach allows a complete control over the processes migration, it implies an additional cost in computational resources as it is applied for every mesoscopic agent registered in the simulation at each LOD update.

5. EXPERIMENTAL EVALUATION

The approach described above has been implemented and evaluated within a proprietary multi-agent simulator. This system is a synthetic environment engine in which each agent has a motivational tree containing predefined attributes, internal variables, emotion and motivations, and can exhibit complex adaptive behaviors. The agent model contains several processes on which our approach can work, such as perceptions, emotions, decision, planning, navigation and interaction with the environment through Smart Objects. Currently, the system can animate up to 20,000 agents driven by more than 20 motivations within a complex environment.

For these experimentations, we tackled the limitations observed with previous work. Thus, we split the representation of the agent model between two main attribute classes: physical and psychological. We simply do the same for the processes. We assign the aggregation thresholds T_1 to the emotional and decisional processes and T_2 to the planning one, such as $T_1 \leq T_2$. In contrast, we give an infinite threshold to the other physical processes such as perception and navigation. The goal is to allow only those working on psychological attributes to migrate to the mesoscopic level, starting with the most abstract and cognitive ones, later followed by the most concrete if $T_1 \neq T_2$. Doing so, the microscopic agents will share their 'minds' through execution of the mesoscopic processes while their bodies will remain at high-resolution in the simulation. This LOD approach tries to reflect the human characteristic of being more sensitive to the physical or visual inconsistencies – wrong trajectories, oscillations, faulty collision avoidance – than the psychological ones.

By preserving the physical parts of the microscopic agents, we hope to solve the spatial inconsistencies observed during previous experimentations. However, maintaining the perception process at the microscopic level means that the perceptions of the mesoscopic agent are an automatic aggregation of those of its microscopic entities. Moreover, the choice of leaving the process in charge of the interactions with the environment at the microscopic level implies that all parts of the mesoscopic agent interacting with a Smart Object will be disaggregated, following

the definition of the disaggregation utility defined above. Such choice leads to an additional cost in computing resources, but is the easiest way to handle interactions here. Indeed, migrating this process to the mesoscopic level would require specific interaction models in the objects themselves, giving them the ability to interact with only a part of a mesoscopic agent. This point is the most important functional difference between macroscopic and mesoscopic simulations.

We use two scenarios that were used previously. The first one takes place in an initially empty subway station, shown in Figure 3, including various objects such as ATMs, ticket vending machines, beverage dispensers and ticket barriers.

Figure 3. View of a part of the test subway station with mesoscopic agents, symbolized by a center linking its associated microscopic entities.

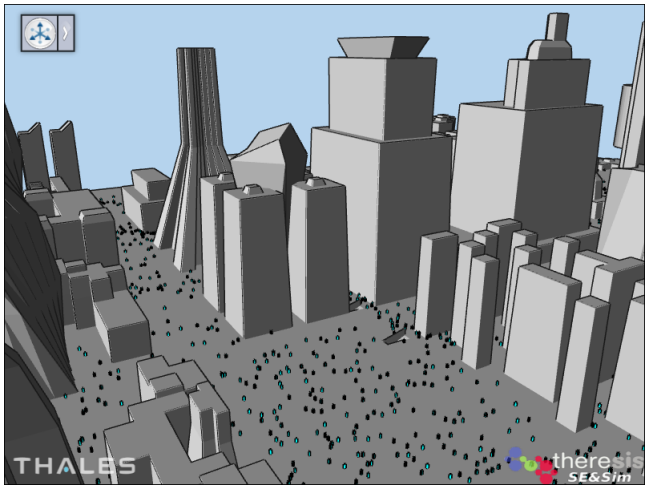

Figure 4. View of the test city.

The second one occurs in a large city, shown in Figure 4, which includes the subway station described above. In each scenario, the agents are animated by a dozen different motivations, such as going to work, drinking, wondering, destroying or repairing a machine, or fleeing. Each incoming agent in the simulation has random internal traits and inventory. We run each scenario with different values for the maximum number of actors allowed in the environment and the maximum size allowed for an aggregate.

Each scenario is run twice – one as a fully microscopic simulation without any LOD process and one with our dynamic aggregation method activated – during 30 minutes on an Intel Core i5 2.50 GHz laptop with a memory of 4 Go. Three criteria are computed: the actual size of the aggregates, the CPU gain and the consistency. The actual size tends to estimate the actual impact of the approach on the simulation and its link with the other two other criteria. The CPU gain is computed by comparing the time needed by both simulations to compute 60 frames. Finally, the consistency is calculated by comparing the cumulative number of uses of each object as a function of time between both simulations. With $U_o(t)$ the cumulative number of uses of object o at time t during the microscopic simulation, and $U'_o(t)$ the cumulative number of uses of the same object at the same time during the LOD simulation, then:

$$Consistency = 100\left(1 - \frac{1}{N_{Objects}} \sum_{o=1}^{N_{Objects}} \left\{\frac{\sum_{t=0}^{T}[U_o(t) - U'_o(t)]}{\sum_{t=0}^{T} U_o(t)}\right\}\right)$$

The subway station has been evaluated with a maximal number of allowed entities ranging from 100 to 1.000, in order to study the impact of this parameter on the criteria described above. Moreover, we set $T_1 = T_2$ in this scenario so that we only have to deal with a single mesoscopic level. In contrast, the number of simulated agents in the city was fixed to 10.000, due to the size of the environment and the time taken by each evaluation run. Unlike previously, we tested two configurations, one with $T_1 = T_2$ and another with $T_1 < T_2$, allowing us to measure the impact of several mesoscopic levels on our evaluation criteria. Finally, the affinity and aggregation utility functions were directly imported from previous work described in [7], such as:

$$Aff(A_1;\ A_2) = \frac{1}{\alpha D_\varphi(A_1;\ A_2)^2 + \beta D_\psi(A_1;\ A_2)^2}, (\alpha;\ \beta) \in {R_+^*}^2$$

$$U_{Ag}(A_1;\ A_2) = \frac{Aff(A_1;\ A_2)}{\gamma D_\varphi(A_1;\ A_2; E)^2}, \gamma \in R_+^*$$

The results of the experimentations done on the subway station are shown in Table 1. It appears that the mesoscopic level allows a slight gain in CPU while the consistency reaches a very high level. Moreover, the real group size is relatively low, regardless of the configured maximum size. As the maximum number of entities in the station increases, the CPU gain decreases and the consistency remains stable. Finally, unlike the simulations with macroscopic aggregates, the strong dissimilarity observed when the maximum number of agents exceeded 500 no longer appears.

This evolution of the criteria can be explained by the preservation of the interaction process at the microscopic level. Indeed, all agents entering the station have at least one interaction with the ticket barriers – and most of them have 2 or 3 more interactions – before reaching a train or an exit. Thus, a lot of disaggregation occurs and the microscopic agents queuing at the machine lower the mean group size as well as the CPU gain. Moreover, only the mental processes were set to migrate to the mesoscopic level, leaving some heavy processes with quadratic complexity, like navigation or perception, at the microscopic level. This explains why the CPU gain is not linear in the actual group size.

Table 1. Experimentation results on the subway stations.

Entities	Max Group Size	Actual Group Size		CPU Gain (%)		Consistency (%)	
		Macro	Meso	Macro	Meso	Macro	Meso
100	5	2,8	2,1	53,1	10,8	98,0	98,8
	10	3,6	2,3	58,3	11,9	97,6	97,2
	15	3,9	2,3	59,9	13,1	95,4	98,1
	20	4,3	2,4	61,2	13,2	92,3	97,6
	25	4,5	2,6	61,7	13,7	91,8	97,3
300	5	3,6	1,7	69,9	7,6	92,6	98,9
	10	4,7	1,9	74,5	9,5	90,7	98,7
	15	5,1	1,9	76,4	9,9	88,0	98,8
	20	5,4	1,9	77,5	10,1	87,4	99,1
	25	5,4	1,9	77,4	10,0	87,4	98,8
500	5	3,5	1,5	71,1	4,8	78,0	98,7
	10	4,0	1,6	74,0	7,0	80,1	98,9
	15	4,4	1,7	76,1	7,5	78,4	98,7
	20	4,6	1,7	76,3	7,6	81,8	98,7
	25	4,7	1,8	76,3	7,8	77,3	98,6
1000	5	3,6	1,1	71,4	2,0	76,3	99,3
	10	4,3	1,3	74,6	3,6	77,8	99,5
	15	4,5	1,2	74,7	2,9	75,4	99,4
	20	4,6	1,3	73,9	3,7	78,1	99,0
	25	4,7	1,4	75,4	3,9	74,7	99,1

Table 2. Experimentation results for the city environment, with $T_1 < T_2$ (Meso 01) and $T_1 = T_2$ (Meso 02).

Max Group Size	Actual Group Size			CPU Gain (%)		
	Macro	Meso 01	Meso 02	Macro	Meso 01	Meso 02
5	5.0	4,9	4,9	69,0	18,2	31,3
10	9,2	9,5	9,4	73,4	24,7	36,4
15	12,5	12,1	12,4	77,7	30,9	40,0
20	13,7	13,4	13,4	81,7	32,8	42,2
25	14,1	13,7	13,9	82,8	33,2	42,5

The impact of the interactions can be observed in the second experimentation. Here, only a few agents among the 10.000 want to take the train in the subway station, the others just walk randomly in the city. Thus, the number of interactions with the objects is smaller than in the first scenario. Table 2 shows that the actual group size is nearly the same for the macroscopic and the mesoscopic scenarios, meaning that the limiting parameter is only the aggregation threshold applied to the aggregation utility. As a consequence, the mesoscopic CPU gain is far higher in this scenario. Moreover, this gain is higher in the experimentation with only one mesoscopic level than in the one with two levels, as expected. It is interesting to note that in this last case, the CPU gain is still significant, suggesting that the emotional and decisional processes require an important amount of computational resources. This result is encouraging as it implies that the approach can save more computational resources in large spaces where agents limit their interactions with the environment.

The comparison between the approach in [7] and the one described here shows that in terms of CPU gain and consistency, the mesoscopic level is an intermediate between the microscopic and macroscopic resolutions. This point is of particular interest here as the mesoscopic level is – by construction – an intermediate toward the construction of the macroscopic one. Thus, if we link the non-mental processes to a threshold whose value is finite and higher than the one defined for the mental processes, then this second aggregation would lead to the creation of a unique macroscopic aggregate as detailed in [7]. The mesoscopic state is then an intermediary step to another resolution level, possibly driven by a different agent model.

Moreover, the stability of the consistency of the mesoscopic level for the simulations involving more than 500 actors, where the macroscopic level shows an important dissimilarity, means that our approach can model the congestions in the station and the evacuation of the agents which are under psychological stress. Indeed, when the subway station is crowded, we see that some agents who cannot access the machines get nervous and leave the station, an interesting and credible event. This phenomenon, which does not exist in the fully macroscopic simulation, remains in the mesoscopic experimentations. Moreover, we observe that the stress of the mesoscopic agent is increasing due to its perceptions, leading it to leave the station. This shows that the interaction between the two resolution levels in the mesoscopic agent leads to consistent actions and can reproduce microscopic behaviors observed in real settings.

6. DISCUSSION AND FUTURE WORK

In this paper, we have presented a novel approach for multi-level agent-based simulations, by introducing an intermediate level between microscopic and macroscopic resolutions. It allows a smoother resolution change between models. Indeed, it supports the definition of several aggregation steps, each corresponding to a process composing the initial agent model, and the migration of the agents to the appropriate aggregation step based on the context of the simulation. It introduces a kind of continuum between the lowest and highest level of simulation.

The results detailed in section 5 show a very high and steady consistency between the fully microscopic and the LOD simulations. On the other hand, the computational gain is not significant in constrained environments where the agents must often interact, but significant when those interactions are less intensive. So, our approach is able to reduce computational needs with no consistency loss as long as the processes maintained at the microscopic level do not need recurrent interactions with their counterpart in other mesoscopic agents, creating partial disaggregation.

This result highlights the importance given to model design in this approach. Indeed, to apply our method, one needs to have a precise view of the available processes, as well as the complete representation of the agent model. While this is always theoretically possible, in practice this may require some modifications of a simulator to control the update of each process and catch the transiting data between them. Moreover, the choice of the aggregation threshold is fundamental as it has a direct impact on the resources, because the processes do not have the same complexity – thus not the same interest to migrate to the mesoscopic level – and because having attributes involved in both microscopic and mesoscopic representations requires the use of the aggregation and disaggregation functions associated with their attribute class. It would be interesting to study the rules that define the optimal aggregation thresholds, depending on the complexity of the processes and their dependency hierarchy. Machine

learning approaches could also help find the best values for the aggregation thresholds.

Finally, it would be particularly interesting to enhance the experimental part of our work. Indeed, by setting more aggregation thresholds, we could test mesoscopic agents having shared perceptions but separate decision and navigation processes, or having only a common long term memory, to test the impact on the consistency. Moreover, we could create enhanced scenarios. Firstly, we could use a train station with a scale larger than our subway station, allowing the agents to have complex behaviors without having too many interactions. We could see if the CPU gain tends to reach the one observed on the city while the consistency remains maximal. Secondly, and more important, we could add more mesoscopic aggregation steps and combine mesoscopic and macroscopic approaches into a unique scenario to verify that the smooth aggregation has an impact on consistency.

7. REFERENCES

[1] Nguyen, T. N. A., Zucker, J. D., Nguyen, H. D., Drogoul, A., and Vo, D. A. 2011. A Hybrid Macro-Micro Pedestrians Evacuation Model to Speed Up Simulation in Road Networks. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*.

[2] Langley, P. and Choi, D. 2006. A Unified Cognitive Architecture for Physical Agents. In *Proceedings of the 21st National Conference on Artificial Intelligence*.

[3] Anderson, J., Bothell, D., Byrne, M., Douglass, S., Lebiere, C., and Qin, Y. 2004. An Integrated Theory of the Mind. *Psychological Review* 111, (4). 1036-1060.

[4] Picault, S. and Mathieu, P. 2011. An Interaction-Oriented Model for Multi-Scale Simulation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*.

[5] Reynolds, P., Natrajan, A., and Srinivasan, S. 1997. Consistency Maintenance in Multiresolution Simulations. *ACM Transactions on Modeling and Computer Simulation*. 7, 3 (July 1997), 368-392.

[6] Hardy, D. and Healy, M. 1994. Constructive & Virtual Interoperation: A Technical Challenge. In *Proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation*.

[7] Navarro, L., Flacher, F., and Corruble, V. 2011. Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*.

[8] Laird, J. 2008. Extending the Soar Cognitive Architecture. In *Proceedings of the 1st Conference on Artificial General Intelligence*.

[9] Davis, P. and Hillestad, R. 1993. Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Management. In *Proceedings of the 25th Winter Simulation Conference*.

[10] Calder, R., Peacock, J., Wise, B., Stanzione, T., Chamberlain, F., and Panagos, J. 1995. Implementation of a Dynamic Aggregation / Disaggregation Process in the JPSD CLCGF. In *Proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representation*.

[11] Karr, C. and Root, E. 1994. Integrating Aggregate and Vehicle Level Simulations. In *Proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation*.

[12] Calder, R., Peacock, J., Panagos, J. and Johnson, T. 1995. Integration of Constructive, Virtual, Live, and Engineering Simulations in the JPSD CLCGF. In *Proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representation*.

[13] Bourrel, E. and Henn, V. 2002. Mixing micro and macro representations of traffic flow: a first theoretical step. In *Proceedings of the 9th Euro Working Group on Transportation Meeting*.

[14] El Hmam, M., Jolly, D., Abouaissa, H., and Benasser, A. Modélisation Hybride du Flux de Trafic. 2005. *Revue électronique Sciences et Technologies de l'Automatique*, 2, (4). http://www.e-sta.see.asso.fr/IMG/pdf/article10-V8.pdf

[15] Rao, D. and Wilsey, P. 2001. Multi resolution network simulations using dynamic component substitution. In *Proceedings of the 9th International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*.

[16] Radhakrishnan, R. and Wilsey, P. 1999. Ruminations on the Implications of Multi-Resolution Modeling on DIS/HLA. In *Proceedings of the 3rd International Workshop on Distributed Simulation and Real Time Applications*.

[17] Stylianou, S., Fyrillas, M., and Chrysanthou, Y. 2004. Scalable Pedestrian Simulation for Virtual Cities. In *Proceedings of the 11th ACM Symposium on Virtual Reality Software and Technology*.

[18] Brom, C., Sery, O., and Poch, T. 2007. Simulation Level of Detail for Virtual Humans. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents*.