Mesoscopic Level: A New Representation Level for Large Scale Agent-Based Simulations

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

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

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

Jean-Daniel Zucker
UPMC – IRD
Paris, France
jean-daniel.zucker@ird.fr

***Abstract* — Large-scale simulations often use multiple models to find a balance between run-time performance and number of simulated entities. Although these approaches are effective, they do not always offer the needed level of analysis, especially when this one is between the resolutions of the models used. In this paper, we aim at offering a finer resolution 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 analysis level. We introduce a framework for such a methodology and evaluate it through the extension of an existing approach along two criteria: its impact on the computational resources, and an estimate of the dissimilarity between a simulation with our methodology and one without. Initial experiments show that the consistency is almost maintained while the CPU gain varies from low to significant depending on the context.**

Keywords – Agent-based and Multiagent Systems; Model-Based Systems; Large-scale simulations; Level of Detail.

I. INTRODUCTION

To understand the complex world surrounding them, humans often resort to models. Models are abstractions of the reality, extracting and simplifying some aspect of the real world to understand its process and predict it. All models are built with a given resolution – or Level Of Detail (LOD) – which depends on the representation level of the entities composing the system and of their behaviors, on the input and output domains and on the scope of the system [1]. The LOD can vary from high-resolution – microscopic – levels, where each component of the system is fully detailed, to lower-resolution – macroscopic – levels that simplify or omit parts of the system to focus only on a behavior of interest. For example, a microscopic description of a gas may be given by the description of each of its particle whereas its macroscopic model may be obtained using the so-called ideal-gas equation law.

Each of those LOD levels addresses different issues and has advantages and drawbacks. High-resolution models allow for a precise grasp of a specific phenomenon and tend to simulate the world more realistically, but they can be very difficult to design due to their high number of parameters. Moreover, their accuracy is often achieved at the cost of huge computational cost. A possible microscopic description of a gas is the position and velocity of each particle as well as the model of their interactions. Considering there are roughly 2.7×10^{19} molecules / cm^3 at room temperature and pressure, it is actually impossible to simulate such a system within a reasonable time. On the other hand, low-resolution models allow a better overall understanding, by focusing on the forest rather than the trees, and are therefore more suitable for decision support. They can be calculated faster but are less accurate and may not fit all situations. Indeed, the ideal gas law requires a few parameters and is far more accessible and usable for engineering applications, although it is restricted to equilibrium states.

Several approaches exist between the molecular model and the ideal gas law, such as Kinetic Molecular Theory, Brownian Motion or high level Boltzmann and Navier–Stokes equations [3]. Each of them is designed for a specific purpose and only works in a particular context – such as low pressure or homogeneity of the molecules – outside of which it is usually inoperative. Thus, there may be questions whose answers lie at the intersection of multiple resolutions, part in the microscopic and part in the macroscopic. In our example, taking care of the particles' motion in a gas while monitoring the whole temperature of a room is impossible because this answer needs both models to be run simultaneously – which cannot be done for the microscopic one. To be obtained, this answer requires a new model.

Those points are particularly important in computer agent-based simulation where the right level of resolution must be found to get the best compromise between the genericity of the system, its intelligibility and its need for CPU and memory resources. In this context, 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 analysis level given the context of the simulation and the user needs. To do so, we first extend the generic notions of dynamic change of representation and spatial aggregation introduced in previous work on the dynamic LOD for agent models [6]. Then, we position ourselves in the context of multi-agent simulations and define several environments in which we evaluate the approach experimentally. Finally we discuss the results obtained and propose enhancements for future work.

In this study, we define several criteria to evaluate the models, such as their scalability – their capacity to simulate a high number of entities – and their precision, thus their ability to give accurate results. We also focus on the design cost needed to implement them as well as the understandability of their results, which is the ease with which the users can understand what is happening.

 ISBN: 978-1-61208-234-9

II. 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 [7]. 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 do not necessarily have hierarchical relationships between them, preventing designers to define which model is the most detailed. Finally, executing the most detailed model does not facilitate decision making.

In aggregation / disaggregation techniques, one model is executed at a given time; but, here it is 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 [5], playboxes [9] and pseudo-disaggregation [8]. 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-levels interactions. Finally, transition latency, network flooding and thrashing may impact simulation consistency.

Variable Resolution Modeling (VRM) allows the creation of families of models that support dynamic changes in resolution [7] 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 [4] 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 loss of resources 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 simultaneous execution of multiple models. But, it only gives mathematical requirements for 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) field also exploit the principle of simultaneous use of microscopic and macroscopic models, by partitioning the environment and running a model in each zone. The pedestrian's simulation described in [13] uses 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, Bourrel and Henn [11] and El Hmam et al. [12] 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 level 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.

III. MESOSCOPIC LEVEL

A. The mesoscopic representation

Our approach start from the foundations defined in [6], where a method to go from several microscopic agents to a single macroscopic aggregate is detailed. Our goal here is to create an intermediate level between the microscopic and the macroscopic ones. Unlike the macroscopic level in which all agents are aggregated into a single one, this 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 define an agent model M as a computational abstraction of the global behavior of a synthetic actor. Thus, it takes as input the representation of the agent being driven and a representation of its environment, and outputs an action or a modification of the agent's representation. This representation – denoted by $Rep(M)$ – is the set of attributes needed by the agent model to perform its task and is usually assimilated to the internal state of the agent. Similarly, the representation of an agent A_1 in M at time t is the vector of attributes' values, denoted $Rep(A_1; M; t)$ such as:

$$Rep(M) = \{a_i\}_{i\in[1..|M|]} \text{ and } Rep(A_1; M; t) = \begin{pmatrix} a_{1;\,1}(t) \\ \vdots \\ a_{1;\,|M|}(t) \end{pmatrix}$$

An agent model can usually be split into several distinct processes, each being a mostly autonomous module leading to a particular skill of the agent such as navigation, decision, emotions, planning, communication or social interactions. For example, in well-known cognitive architectures such as SOAR and ACT-R [2], 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

 ISBN: 978-1-61208-234-9

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. For example, in SOAR, the 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.

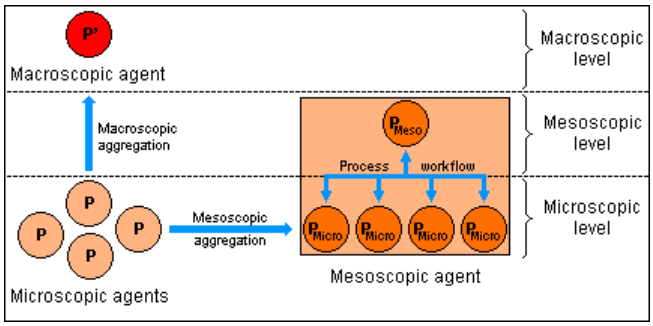

Figure 1. Example of the macroscopic and mesoscopic aggregation of 4 agents implemented as a set of P processes.

We consider that the agent model M is composed of a set of m processes $P = \{P_1;\ P_2;\ \ldots;\ P_m\}$. The goal, schematized 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 n agents $A = \{A_1;\ A_2;\ \ldots;\ A_n\}$ in $P_k \in P_{meso}$ at time t are respectively denoted by $Rep_{micro}(A_1; P_k; t)$ and $Rep_{micro}(A; P_k; t)$ such as:

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

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

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 [6]. Firstly, we partition the representation of the agents in M among several attributes classes in which each attribute share the same meaning and therefore a common dynamics. Then, we link each class with an aggregation operator and its corresponding disaggregation and memory operators. Then, we are able to define aggregation, disaggregation and memory functions, respectively denoted by F_{Ag}, F_{Disag} and F_{Mem} and for the process P_k such as:

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

$$F_{Mem}[Rep_{meso}(A; P_k; t); Rep_{micro}(A;\ P_k; t)] = Mem(A;\ P_k; t) = \begin{pmatrix} m^k_{1;1}(t) & \cdots & m^k_{n;1}(t) \\ \vdots & \ddots & \vdots \\ m^k_{1;|P_k|}(t) & \cdots & m^k_{n;|P_k|}(t) \end{pmatrix}$$

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

This method allows a single process to work at the mesoscopic level. 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 a is only used at the microscopic level, then it is ignored. On the other hand, if a is only used at the mesoscopic level, meaning that $a \notin \bigcup_{P_k \in P_{micro}} Rep(P_k)$, it is aggregated once. Finally, if a is used at both level, we need to maintain both microscopic and mesoscopic values of a when it is updated by any process, with the aggregation, disaggregation and memory functions described above for the attributes class to witch a belong. 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.

Such approach permits 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 [7]. The choice of the process to migrate is also an important issue. It is better to migrate processes that require high computation resources, and it is easier to migrate those that have few dependencies with others. However, the choice of the processes to migrate has also an impact on the simulation consistency. Finally, it is important to notice that migrating all processes from the microscopic to the mesoscopic level is equivalent to aggregating the set of agents to a macroscopic one driven by the same agent model as defined in [6].

B. *Spatial aggregation*

This section tackles the problem of finding the agents that should be aggregated to form a mesoscopic agent, and which processes of this new agent must be migrated to the mesoscopic level. The philosophy employed here is similar to the one detailed in [6] with the definition of a spatial distance D_Θ, and a psychological distance D_Ψ combined to estimate an affinity between two agents A_1 and A_2, denoted by $Aff(A_1; A_2)$, and the affinity between two agents A_1 and A_2 and a set of M events $E = \{E_1; E_2; \ldots; E_M\}$, denoted by $Aff(A_1; A_2; E)$, such as:

$$Aff(A_1; A_2) = f[D_\Theta(A_1; A_2); D_\Psi(A_1; A_2)]$$

$$Aff(A_1; A_2; E) = \max_{i \in [\![1;|E|]\!]} \left[f[D_\Theta(A_1;\ A_2;\ E_i); D_\Psi(A_1;\ A_2;\ E_i)]\right]$$

$$with \begin{cases} D_\Theta(A_1; A_2;\ E_i) = \mathrm{Min}[D_\Theta(A_1; E_i);\ D_\Theta(A_2; E_i)] \\ D_\Psi(A_1; A_2; E_i) = \mathrm{Min}[D_\Psi(A_1; E_i);\ D_\Psi(A_2; E_i)] \end{cases}$$

However, the definition of D_Θ and D_Ψ is not trivial because parts of the physical or psychological processes may be at the microscopic or the mesoscopic level. In [6], the affinities are combined to define the aggregation utility U_{Ag} between two agents, which represents the usefulness of creating a mesoscopic entity considering those agents and the simulation context. This functionality is enhanced here by the ability to choose which process should migrate to the mesoscopic level while the others stay at the microscopic one. To do so, we define for each process an aggregation threshold, thus creating a total order over them, meaning that the lower the threshold, the more likely the process is to be migrated to the mesoscopic level. Those thresholds do not depend 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_Θ and D_Ψ. Although this approach requires significant computational resources, it allows disaggregating a single microscopic agent due to the simulation context, which was not possible with the macroscopic aggregate. 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 allow the processes to migrate dynamically when the mesoscopic agent is alone. However, it is possible to define a representation change utility for a mesoscopic (or macroscopic) agent A, denoted by $U_{RC}(A)$, such as:

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

The representation change utility 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 computation level of its own processes. While this approach allows a complete control over the processes migration, it may require additional CPU resources if it is applied at each LOD update for every single agent.

IV. EXPERIMENTAL EVALUATION

The approach described above has been implemented and evaluated within a Thales proprietary multi-agent simulator named SE-*. 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, SE-* can animate up to 20,000 agents driven by more than 20 motivations within a complex environment.

For these experimentations, and to keep a common context with [6], we split the representation of the agent model between two main attributes classes: physical and psychological. We do the same for the processes, thus linking the mental processes – emotions, decision and planning – to a unique aggregation threshold while the other physical processes – perception and navigation – are assigned an infinite threshold. The goal is to allow only the processes working on psychological attributes to migrate at the mesoscopic level. Doing so, the microscopic agents will share their thoughts through the mesoscopic one while their bodies will remain on the simulation. This LOD approach tries to reflect the human ability to be more sensitive to the physical or visual inconsistencies – wrong trajectories, oscillations, bad avoidance – than the psychological ones.

By keeping the physical parts of the microscopic agents, we hope to solve the spatial inconsistencies observed during the experimentations in [6]. Thus, letting the perception process at the microscopic level means that the perception of the mesoscopic agent are an automatic aggregation of those of its microscopic entities. Moreover, the choice of leaving the process manage 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 a 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 already defined in [6]. The first one takes place in a subway station initially empty, including various objects such as ATMs, ticket vending machines, beverage dispensers and ticket barriers. The

 ISBN: 978-1-61208-234-9

second one occurs in a large city that incorporates the subway station described above. In each scenario, the agents are driven 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.

As in [6], 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 GB. 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 to link it with the 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 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 exceeds 500 no longer exists.

TABLE 1. EXPERIMENTATION RESULTS ON THE SUBWAY STATIONS.

Entities	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
	20	4,3	2,4	61,2	13,2	92,3	97,6
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
	20	5,4	1,9	77,5	10,1	87,4	99,1
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
	20	4,6	1,7	76,3	7,6	81,8	98,7
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
	20	4,6	1,3	73,9	3,7	78,1	99,0

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.

The impact of the interactions can be observed in the second experimentation. Here, only a few agents among the 10000 want to take the train in the subway station, the others just walk randomly in the city. Thus, the number of interactions with 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 scenario, 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. 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.

TABLE 2. EXPERIMENTATION RESULTS FOR THE CITY ENVIRONMENT.

Entities	Group Size	Actual Group Size		CPU Gain (%)	
		Macro	Meso	Macro	Meso
10000	5	5.0	4,9	69,0	31,3
	10	9,2	9,4	73,4	36,4
	20	13,7	13,4	81,7	42,2

The comparison between the approach in [6] 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 second aggregation threshold which value is 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 [6]. 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 agents that cannot access the machines getting nervous and leaving the station. This phenomenon, which does not exist in the fully macroscopic simulation, remains in the mesoscopic experimentations.

 ISBN: 978-1-61208-234-9

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 both resolution levels in the mesoscopic agent leads to consistent actions and can reproduce microscopic behaviors observed in real settings.

V. DISCUSSION AND FUTURE WORK

In this paper, we presented a novel approach for multi-levels simulations, by introducing an intermediate level between microscopic and macroscopic resolutions, which 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 in regard to the context of the simulation. It corresponds to a kind of continuum between the lowest and highest level of simulation.

The results detailed in Section IV 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 point 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. Thus, by setting different 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 larger scale 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.

REFERENCES

[1] J. D. Zucker. *A grounded theory of abstraction in artificial intelligence*. Philosophical Transactions of the Royal Society (volume 358, number 1435). 2003.

[2] P. Langley, J. Laird and S. Rogers. *Cognitive Architectures: Research Issues and Challenges*. Cognitive Systems Research. 2009.

[3] J. Uffink. *Compendium of the foundations of classical statistical physics*. Philosophy of Physics (volume 2, number 1). 2006.

[4] P. Reynolds and A. Natrajan. *Consistency Maintenance in Multiresolution Simulations*. ACM Transactions on Modeling and Computer Simulation. 1997.

[5] D. Hardy and M. Healy. *Constructive & Virtual Interoperation: A Technical Challenge*. In proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation (CGF-BR 1994). 1994.

[6] L. Navarro, F. Flacher and V. Corruble. *Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations*. In proceedings of the 10th International Confer-ence on Autonomous Agents and Multiagent Systems (AAMAS'11). 2011.

[7] P. Davis and R. Hillestad. *Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Management*. In proceedings of the 25th Winter Simulation Conference (WSC'93). 1993.

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

[9] C. Karr and E. Root. *Integrating Aggregate and Vehicle Level Simulations*. In pro-ceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation (CGF-BR 1994). 1994.

[10] Calder, J. Peacock, J. Panagos and T. Johnson. *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 (CGF-BR 1995). 1995.

[11] E. Bourrel and V. Henn. *Mixing micro and macro representations of traffic flow: a first theoretical step*. In proceedings of the 9th Euro Working Group on Transpor-tation Meeting (EWGT2002). 2002.

[12] M. El Hmam, D. Jolly, H. Abouaissa and A. Benasser. *Modélisation Hybride du Flux de Trafic*. Revue électronique Sciences et Technologies de l'Automatique. 2006.

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

 ISBN: 978-1-61208-234-9