Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations

Laurent Navarro
Thales / UPMC – LIP6
4 place Jussieu
75005 Paris, France
laurent.navarro@lip6.fr

Fabien Flacher
Thales
20-22 rue Grange Dame Rose
78140 Vélizy, France
fabien.flacher@thalesgroup.com

Vincent Corruble
UPMC – LIP6
4 place Jussieu
75005 Paris, France
vincent.corruble@lip6.fr

ABSTRACT

Large scale agent-based simulations typically face a trade-off between the level of detail in the representation of each agent and the scalability seen as the number of agents that can be simulated with the computing resources available. In this paper, we aim at bypassing this trade-off by considering that the level of detail is itself a parameter that can be adapted automatically and dynamically during the simulation, taking into account elements such as user focus, or specific events. We introduce a framework for such a methodology, and detail its deployment within an existing simulator dedicated to the simulation of urban infrastructures. We evaluate the approach experimentally along two criteria: (1) the impact of our methodology on the resources (CPU use), and (2) an estimate of the dissimilarity between the two modes of simulation, i.e. with and without applying our methodology. Initial experiments show that a major gain in CPU time can be obtained for a very limited loss of consistency.

Categories and Subject Descriptors

D.3.3 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Multiagent systems*

General Terms

Algorithms, , Performance, Experimentation.

Keywords

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

1. INTRODUCTION

Agent-based simulation of credible actors in large-scale urban environments is a growing research domain, with numerous applications ranging from security to crisis management, entertainment, urban planning and virtual training. Those simulations share broadly speaking the same high-level goal: provide a powerful analytical tool which can animate a large number of individuals, with complex, credible – sometimes realistic – behavior, within a large world. Ideally, they would work in real time in a continuous space, on a standard machine and with intensive and rich interactions with one or several users.

However, simulating hundreds of thousands of individual agents within a very large environment like an airport, a crowded train station or a whole megacity, with credible behaviors, requires important computational power. This is mainly due to the complexity of the microscopic models used for instance for navigation, or decision processes that result in large states and actions spaces. Indeed, most of them require that each agent perceive its environment, update its internal variables, choose the most appropriate action and eventually communicate and learn. Reducing the complexity of the underlying algorithms is then a significant challenge.

A similar issue has already been tackled by the field of computer graphics, where Level of Detail (LOD) techniques have been investigated [1] in order to find a good balance between visual credibility and computational requirements. Those techniques tend to adapt the complexity of the 3D models based on the viewpoint of the observer. Our approach proposes a similar idea adapted to the agent models.

In this paper, we define an agent model as a computational abstraction of the behavior or the cognitive capabilities of a synthetic actor. Thus, this definition either applies to the processes and behaviors dealing with navigation, decisions, emotions, communication or social interactions. All those models take as input a representation of the agent being driven and a representation of its environment, and output an action or a modification of the internal state.

We present here a novel approach of dynamic LOD for large scale simulations, which can apply to all agent models. Moreover, instead of using predefined LOD levels, our approach is able to determine by itself the most suitable representation level for each agent, regarding the simulation context, in real time and within a continuous environment. To do so, we first introduce the generic notions of dynamic change of representation and spatial aggregation. Then, we define a concrete sub-problem and we evaluate the approach experimentally along two criteria: the impact of our methodology on the computational resources, and an estimate of the dissimilarity between a full microscopic simulation and a simulation with our methodology. Finally we discuss the results obtained and propose enhancements for future works.

2. RELATED WORK

Generating realistic behavior for virtual humans has been the subject of numerous studies in various communities. Systems like SOAR [11], ACT-R [3], ICARUS [2] or LIDA [19] are excellent examples of cognitive architectures that provide a complex modeling of extremely advanced human reasoning capabilities at microscopic scale, based on studies about human memory, problem solving and skill acquisition [4]. However, though these systems are applicable in scenes with a reasonable number of

Cite as: Dynamic Level of Detail for Large Scale Agent-Based Urban Simulations, L. Navarro, F. Flacher and V. Corruble, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 701 - 708.

actors, they are inefficient to handle applications involving large populations of virtual humans on a standard computer. This limitation is also a disadvantage to the use of multi-agent platforms such as Cougaar [8], JADE [17] and ZEUS [26] which offer specific architectures able to distribute the virtual entities on different machines depending on the required computational load.

Attempts have been made to increase the number of simulated entities on a single computer by tuning the update time length given to each agent. To reach the amount of 200.000 vehicles simulated as individual autonomous agents with specific action selection mechanisms, SUMO [23] uses discrete calculation time steps of 1 second. Similarly, the crowd simulation proposed in [18] reduces the update times of non-visible agents and adapts their behavior to more simplified but less accurate microscopic agent models. Finally, the Process Manager described in [10] dynamically chooses between several AI update processes – full, time-sliced, postponed or replaced with simplified behavior – depending on the needs in computational resources. While those systems share the same philosophy, the first one sacrifices its real-time component for the benefit of an accurate result whereas the others elected to decrease the realism of the simulation to maintain its believability.

Some systems are able to simulate a very large number of agents using only macroscopic models. Crowd Patches [9] can handle up to 3.700 actors by dividing the world into small convex areas where agents can navigate, and using offline computed paths and animations stored within each patch to steer them. Other approaches have been attempted through the simultaneous use of macroscopic and microscopic models to define the individual behaviors of each agent. Thus, YaQ [25] uses offline predefined macroscopic paths across the world to steer up to 35.000 pedestrians using various microscopic algorithms, depending on their position: potential fields on significant areas, Craig Reynolds's seeking behavior on lower interest spots and linear steering toward their destination without collisions on unimportant regions. Similarly, Continuum Crowds [7] represents agents as particles which are subjected to three fields – one for their destination, one for their speed and one for their discomfort caused by the proximity of other agents – that guide them to their destination. Thereby, those systems combine global path planning and local collision avoidance within a single global steering model. However, they focus on navigation issues and are not easily transposed to other levels of behavior models such as ones dealing with decisions or emotions. Moreover, they do not provide the expected level of interactivity.

Some approaches also exploit the principle of simultaneous use of microscopic and macroscopic models, but choose to partition the environment and implement a model type for each zone. [22] describes a top-down approach for simulating pedestrians within a large city, which uses high level flows and distributions 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, the systems presented in [20] and [21] simulate vehicles navigating in a static predesigned world. The entities use a macroscopic model based on the flow theory for low interest areas without crossroads, and a microscopic multi-agent car-following model for 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.

A last approach, IVE [16], is of particular interest to our work, since it is one that introduces level of detail techniques on human decision and behavior. This framework utilizes a hierarchical reactive planning mechanism to control the agents, which uses a tree structure. Those agents are placed within a 2D world that is split into atomic cells which are hierarchically organized within a topology tree. Each level of this topology tree is linked to one of the behavioral tree, defining accessible LOD ranks. Thus, IVE can adapt the level of detail of the simulation in order to simplify the behaviors of the unobserved agents – and then reduce the computational needs – hence dealing with more than 10.000 agents simultaneously. But it requires the use of a discrete hierarchical world statically linked with the tree structure used by the decision process.

The field of multi-agent systems is not the only one to be relevant in the context of this study. Thus, Multi-Resolution Modeling (MRM), which is the joint execution of different models of the same phenomenon within the same simulation or across several heterogeneous systems, provides several relevant approaches. In selective viewing [12], only the most detailed model is executed, and all other ones are emulated by selecting information, or views, from the representation of the most detailed model. In aggregation / disaggregation techniques, one model is executed at a given time, but instead of being the most detailed one like in selective viewing, the choice of the model depends on the user needs. This approach has several variants, such as full disaggregation [15], partial disaggregation [6], playboxes [14] and pseudo-disaggregation [13]. Variable Resolution Modeling allows the construction of families of models which support dynamic changes in resolution [12] by introducing constraints during their creation, such as the standardization of all the parameters in a dictionary, the creation of a hierarchical structure for the variables or the definition of calibration rules between models.

Multiple Representation Entities [5] is a final example from the MRM field which is of particular interest here. It uses concurrent representations to ensure simulation consistency and reduce computation costs. Its approach is to maintain, at all time, all representations through all available models of a given phenomenon, using appropriate mapping functions to translate changes between two representations. The goal is to permit constant interactions between all the representations, to avoid loss of resources or time when scaling from one model to another. This approach 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.

3. DYNAMIC LEVEL OF DETAIL FOR AGENT MODELS

Our approach aims to mix the philosophy of graphical level of detail with the use of multiple agent models at different resolutions. The goal is to simulate precisely the behavior of actors in areas of high level of interest with microscopic models and to simulate less precisely but more economically (resource-wise) behavior of actors located elsewhere with macroscopic models.

Several criteria have motivated the choice of using multiple models. Firstly, it allows the capture of all the aspects of a given phenomenon. Indeed, low resolution models allow a better overall understanding, by focusing on the big picture rather than on the details, whereas high resolution models give an accurate comprehension of a specific phenomenon and tend to simulate reality. Secondly, such a choice allows the finding of a good balance between computing resources and simulation properties, such as realism, coherence and complexity. Indeed, although high resolution models are very accurate for modeling individual behaviors, they often have high computational and memory needs. On the other hand, low resolution models can save resources but tend to give less accurate results. Mixing both types of models can hopefully lead to the best of both worlds. Finally, using multi models helps design systems by mimicking the human reasoning ability – which already works at different levels of understanding – and simplifies the calibration of the models by allowing the use of available data matching at least one of the implemented models.

However, this fundamental choice leads to several challenges which can be classified along two axes. The first one relates to the models themselves. One must define the way they will be used (a model at a time, one model per areas of interest, all models simultaneously, etc…) and the way they will interact, using some of the Multi Resolution Modeling methods described above. The second axis relates to the physical agents. One must define how to manage a continuous 3D environment with complex moving agents, and how the physical position of the agents will have an impact on the model used.

3.1 Dynamic change of representation

This chapter focuses on the scalability aspect of the implemented agent models. It attempts to provide an efficient method for navigating dynamically from one model to another. The primary decision made is the choice of the aggregation / disaggregation technique to define how the models are used. This way, several agents are aggregated into a group of agents, then several groups are aggregated into a crowd, and finally several crowds are aggregated into a flow. The different agent models (agent, group, crowd and flow) are linked to each aggregation / disaggregation step.

Let M_1 be an agent model. The representation of an agent A_1 in M_1 at time t is denoted by $Rep(A_1;\ M_1;t)$ and is the vector of inner attributes of A_1 required by M_1 to operate. The number of such attributes is denoted by $|M_1|$. Then:

$$Rep(A_1;\ M_1;t) = \begin{pmatrix} a_{1;\,1}(t) \\ a_{1;\,2}(t) \\ \vdots \\ a_{1;\,|M_1|}(t) \end{pmatrix}$$

Let M_2 be another agent model. We assume that M_2 is more abstract than M_1, which also means that the representation level of M_1 is higher than the one of M_2. Finally, let $A = \{A_1;\ A_2;\ \dots;\ A_N\}$ be a set of N agents, driven by the model M_1. The goal is to find the aggregation function F_{Ag} able to transform the representation of A in M_1 at time t, into the representation of the aggregate A' controlled by the model M_2 at the same time:

$$\begin{aligned} Rep(A;\ M_1;t) &= \big(Rep(A_1;\ M_1;t);\dots;\ Rep(A_N;\ M_1;t)\big) \\ &= \begin{pmatrix} a_{1;\,1}(t) & \cdots & a_{N;\,1}(t) \\ \vdots & \ddots & \vdots \\ a_{1;\,|M_1|}(t) & \cdots & a_{N;\,|M_1|}(t) \end{pmatrix} \end{aligned}$$

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

As is, such function is difficult to define – or to learn – because it attempts to aggregate parameters which are a priori not semantically connected, such as the velocity of the agents and their thirst level. Our approach is to split F_{Ag} into several sub functions, each operating on parameters with a similar meaning, therefore likely to share a common dynamic. In this end, we classify each agent's attributes in two categories, physical and psychological, and several subcategories, like physical traits, resources or spatial data for the first group and emotions, internal variables or knowledge for the second. Then, we partition the representation of the agents in each model. The goal is then to find the aggregation sub functions corresponding to each class of attributes, which guarantees the consistency of the models and allows a future disaggregation.

The notion of consistency is central in such an approach because it symbolizes the amount of essential information lost during the aggregation / disaggregation process and is linked to the global coherence of the simulation. A relevant definition of consistency between a high level model M and a low level model M' has been given in [12] by the comparison between the projected state of an aggregate of high level entities which have followed M, and the projected state of the same aggregate initially controlled by M'. The projection symbolizes that only a part of the final states is relevant to define the consistency. Our approach uses this notion to determine which kind of sub function fits best with which class of attributes. Thus, machine learning techniques would allow the system to find the best sub function for each attributes class between two agent models among a group of predefined operators such as SUM, MIN, MAX, MEDIAN or MEAN, by optimizing the consistency of both models.

In parallel to the definition of the aggregation sub functions, we must find the associated disaggregation operator, F_{Disag}, which aims to recreate A from A' at time t' with respect to the evolution of A' between t and t'. To do so, we define memory functions whose goal is to save data at aggregation time to facilitate the disaggregation process:

$$\begin{aligned} Mem(A;\ M_1;t) &= \big(Mem(A_1;\ M_1;t);\dots;\ Mem(A_N;\ M_1;t)\big) \\ &= \begin{pmatrix} m_{1;\,1}(t) & \cdots & m_{N;\,1}(t) \\ \vdots & \ddots & \vdots \\ m_{1;\,|M_1|}(t) & \cdots & m_{N;\,|M_1|}(t) \end{pmatrix} \end{aligned}$$

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

There is a strong link between an aggregation function, its opposite disaggregation operator and the associated memory function. As an example, let us consider the resources of an agent. An intuitive aggregation operator would be the SUM as we may consider that a group of agents disposes of the sum of the resources of each individual. In this case, the memory function would be, for each resource attribute, a RATIO operator between the initial amount of the aggregated agent and the amount of the aggregate. Then, the disaggregation function would be a simple MULTIPLY between the new amount of the aggregate and the memory of the agent, plus a random distribution of surplus between the agents.

Finally, such method allows our approach to tune the memory consumption by controlling the quantity of data stored by the memory functions for each aggregated agents. Thus, gradual forgetting methods can be implemented, which keeps all the data of $Mem(A;\ M_1; t)$ just after the aggregation, then creates a statistical distribution for each attribute among all the aggregated agents after a predefined period of time and finally erase all stored data if the agents have been aggregated after a long period. In this last case, random attributes are generated for the disaggregation process.

3.2 Spatial aggregation

This section focuses on the spatial aggregation of agents and addresses the issue of finding which agents should be aggregated to form a representation at a less detailed level. The philosophy employed here is to consider a group of humans as a set of individuals with similar psychological profiles and a common physical space.

To this end, two distances are defined based on the two main attributes classes defined before: a spatial distance D_Θ, and a psychological distance D_Ψ. The first one can be a trivial Euclidean distance or a more complex computation taking into account the physical path between the two agents. The second distance represents how two actors share the same thoughts (for example the same goal, the same dominant emotion or the same desire). It can be the norm between the vectors of psychological attributes or the similarity between the long term goals chosen by the agent. Those distances are combined to define the affinity between two agents A_1 and A_2.

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

This affinity must be a continuous positive function, strictly decreasing as D_Θ or D_Ψ increase. It represents the connection between two agents within the simulation, only based on their individual states. Their environment is taken into account with the definition of events. Those symbolize points of particular attention which require the creation of an area of high level of interest to increase the overall consistency of the simulation. Thus, the observer's point of view, an accident or an evacuation can lead to the creation of simulation events. Let $E = \{E_1;\ E_2;\ \ldots;\ E_M\}$ be a set of M events generated by the simulation. The link between an agent and an event is characterized by a new pair of distances similar to those defined above. Although the meaning of the physical distance remains the same as the one between two agents, the signification of the psychological one is a bit different, and symbolizes how an actor is sensitive to the event. For example, if we consider an agent collapsing in the street, we can assume the impact of this event to be higher on a doctor walking nearby than on a child or an employee in a hurry. Those distances are combined to define the affinity between two agents A_1 and A_2 and an event $E_i \in E$:

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

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

Finally, we can define the link between the two agents A_1 and A_2 and E:

$$Aff(A_1;\ A_2;\ E) = \underset{i\in[1;M]}{\mathrm{Max}}[Aff(A_1;\ A_2;\ E_i)]$$

This link is finally used to define the aggregation utility between two agents A_1 and A_2. This utility guides the choice of which agents to aggregate because they are close in their representation space and are not of interest for the simulation.

$$U_{Ag}(A_1;\ A_2) = f[Aff(A_1;\ A_2);\ Aff(A_1;\ A_2;\ E)]$$

The computation of the aggregation utilities between the agents leads to the creation of an aggregation graph, which vertices are the agents in the simulation. An edge of the graph is created when the value of the aggregation utility is greater than a given threshold. The weight of the edge is set to the value of the utility. Figure 1.A shows agents symbolized by circles with different colors representing their psychological states. The corresponding graph is shown in Figure 1.B. This structure allows optimizing the repartition of the agents within the created groups – Figure 1.C – with the use of specific graph algorithms.

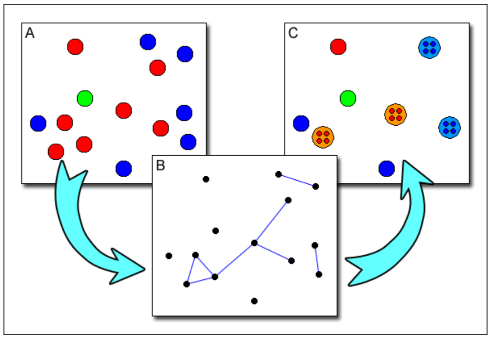

Figure 1: Example of spatial aggregation, with agents on the ground (A) used to create an aggregation graph (B) finally leading to the formation of groups (C).

The disaggregation of an aggregate A' proceeds of the same idea, although it just take into accounts the events defined in the simulation. Thus, we can define an affinity between A' and an event $E_i \in E$, then the affinity between A' and E, and finally the disaggregation utility which guides the choice of which aggregate to split because its representation grain is too coarse for the area of interest where it stands:

$$Aff(A';\ E_i) = f[D_\Theta(A';\ E_i); D_\Psi(A';\ E_i)]$$

$$Aff(A';\ E) = \underset{i\in[1;M]}{\mathrm{Max}}[Aff(A';\ E_i)]$$

$$U_{Disag}(A') = f[Aff(A'; E)]$$

3.3 Implementation in SE-*

A large part of our approach has been implemented and evaluated within SE-*, a Thales proprietary multi-agent simulation. This system is a synthetic environment engine, designed to be highly scalable and capable of modeling complex adaptive behaviors, low-level navigation and interactions with the environment. Each agent has a motivational tree containing predefined attributes, internal variables, motivations and behaviors. A hierarchical plan is created from these different motivations and from the Smart Objects the agent may use. Currently, SE-* can animate up to 20,000 agents driven by more than 20 motivations within a complex environment.

This simulator has been used to test our approach on several scenarios. Due to the complexity of the model described above and its large number of parameters, we decided to focus on a sub problem for this first experiment, mostly by reducing the scalability of the agent models. The main simplification is the definition of two representation levels – individual and group – and the use of the same microscopic navigation and decision model for both levels. Thus, we assume that a group of a small number of agents perceive and act like a single actor. Then, we classify the attributes of the model into 3 physical categories (physical traits, resources and spatial data) and 3 psychological ones (motivations, internal variables and psychological traits). Finally, because our approach does not implement yet any automated learning mechanism for finding the aggregation operators, we defined them by hand. Thus, we use a simple MEAN operator for all the categories except for the resources which are aggregated using a SUM operator. The associated disaggregation and memory operators were also designed by hand.

To compute the affinity between two agents A_1 and A_2, we implemented a basic Euclidian distance as D_Θ and we set D_Ψ as being equal to zero if the agents have the same short-term goal, one if not. The affinity function is then defined as follow:

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

The affinity between two agents A_1 and A_2 and an event $E_i \in E$ is defined similarly, except for D_Ψ which is always zero, symbolizing the fact that the agents are always affected by the events of the simulation. The aggregation utility between two agents A_1 and A_2 is then defined as follows:

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

Considering that D_Ψ is always zero, the definition of the disaggregation utility for an aggregate A' proceeds of the same idea:

$$U_{Disag}(A') = Aff(A'; E) = \frac{1}{\delta D_\Theta(A'; E)^2}, \delta \in R_+^*$$

4. EXPERIMENTAL EVALUATION

We designed 3 scenarios to evaluate our approach. Two of them take place in a subway station initially empty, including various objects such as ATMs, ticket vending machines, beverage dispensers and ticket barriers, and the last one occurs in a large city. In each scenario, the agents are driven by a dozen different motivations, such as going to work, drinking, destroying a machine, repairing a broken machine or fleeing.

Two subway stations have been designed for the two first scenarios, which share the same 3D model but have specific locations for the objects. Details are shown in Figure 2 and Figure 3. When entering the station, each agent aims to take the train and has random physical and psychological traits as well as 30% chance to own a ticket and another 30% chance to start with a small amount of money. To achieve its initial goal, and according to its inner attributes, an agent will have to get some cash at the ATM, buy a ticket, get a drink or directly go through the ticket barriers to the train doors. The first station contains 4 entries, 4 train doors, 8 ATMs (in green on the figures), 8 ticket vending machines (in yellow), 12 ticket barriers (in white), 12 exit barriers (in dark red) and 7 beverage dispensers (in red). In the second one, 4 ATMs were swapped with 4 ticket vending machines in order to see if a modification in the topology has an impact on the performances.

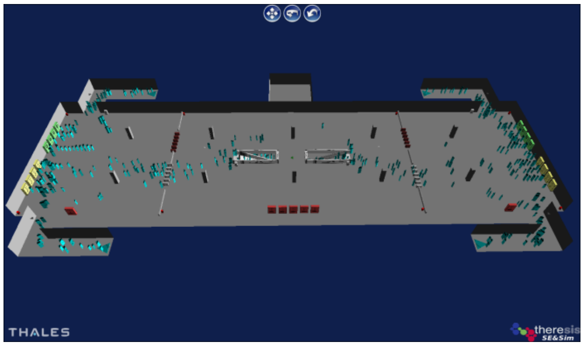

Figure 2: Top view of the first test subway station.

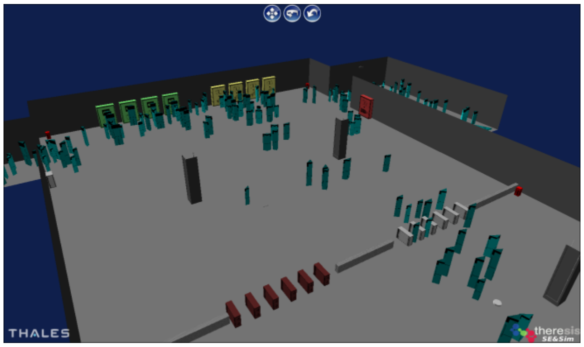

Figure 3: View of a part of the first test subway station.

The last scenario takes place in an entire city which includes the subway station, shown in Figure 4. The 3D mesh is larger and allows the simulation of thousands of agents. However, it does not contain any smart objects with which to interact. Thus, the agents only walk from entry points to exit gates without colliding, which is a typical navigation task.

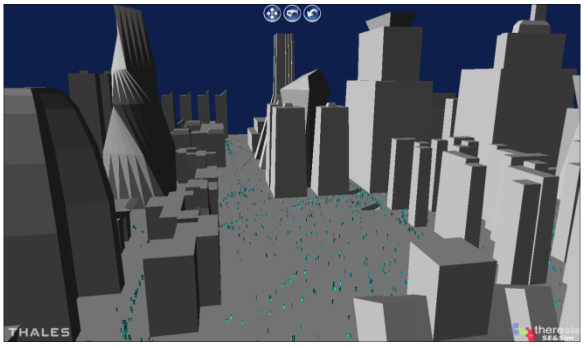

Figure 4: View of the test city.

Each scenario was run twice – one as a fully microscopic simulation without any LOD process and one with our dynamic aggregation method activated. The goal was to compare both runs to calculate both the CPU gain and the behavioral consistency. For the first criterion, we stored the total amount of time needed by the simulation to compute 60 frames within one second. For the second one, we aimed to find an estimate of the behavioral distance between both runs. Thus, we used as objective abstract criterion: the number of uses of each object, from the start of the simulation to the measure time. Those cumulative values, taken every second, symbolize the throughput of each machine within the station. Because we already assume that the aggregation process has an impact on the simulation that is unavoidable and that may be significant, we choose to avoid using exact statistical hypothesis tests, such as Mann-Whitney's. Instead we defined, for each object, a local dissimilarity as the difference of the temporal means between the cumulative values obtained at both runs. Finally, we defined a global behavioral dissimilarity indicator as the mean of all the variations found for all objects. Let $U_o(t)$ be the cumulative number of uses of object o at time t during the microscopic simulation, and $U'_o(t)$ the cumulative uses of the same object at the same time during the simulation using our dynamic aggregation method. Then:

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

Because it does not have any smart object to interact with, only the CPU gain was computed for the city scenario. For each scenario, we changed the maximum number of agents within the simulation and the maximum number of entities allowed inside an aggregate in order to study the impact of those parameters on the results. Finally, each experimentation has been run 5 times during 30 minutes on an Intel Core 2 Duo 2.26 GHz laptop with a memory of 2 Go. The results showed are the mean of the 5 runs.

Table 1: Experimentation results on both subway stations varying max group size and max number of entities.

Entities	Max Group Size	CPU Gain (%)		Dissimilarity (%)	
		1st Station	2nd Station	1st Station	2nd Station
100	5	43,4	43,0	3	5,4
100	10	47,5	45,9	7,1	6,9
100	15	50,3	46,5	10,9	8,3
100	20	49,4	46,9	9,9	7,6
100	25	50,5	47,6	8,7	9,7
300	5	59,9	56,7	4,9	6,3
300	10	66,7	60	4,5	5,5
300	15	67,9	65,6	7,5	8,5
300	20	67,7	66,2	5,7	6,3
300	25	69	66,9	8,6	7,1
500	5	61,5	56,8	21,5	20,1
500	10	67,4	64,5	19	19,1
500	15	69,6	67,2	18,7	18,4
500	20	70,7	66,5	17,2	17,5
500	25	72,6	69,1	14,2	16,2
1000	5	57,33	53,8	35,41	36,1
1000	10	63,97	59,4	33,68	32,4
1000	15	66,52	58,7	33,85	32,3
1000	20	67,79	60,7	31,51	31,4
1000	25	68,79	61,3	32,6	31,4

The results of the experimentation done on the first station are shown in Table 1. It appears that, for a given maximum number of agents within the station, the CPU gain is very encouraging (between 40% and 70% is saved) and logically increases with the maximum size of each aggregate. On the other hand, the behavioral dissimilarity appears to be acceptable (3-10% range for simulation inconsistency) for a maximum of 100 and 300 agents in the station. However, it becomes unsatisfactory (14 to 36% inconsistency) if the station is filled with 500 or 1000 agents. Moreover, there is no clear pattern in the dynamics of the behavioral dissimilarity as a function of the group size.

Table 1 also shows the results obtained when running the tests on the second station. The evolution of the CPU gain is the same as the one observed in the first experiment. However, the behavioral dissimilarity seems to be globally better at 300 agents even if it remains in the same range. Like before, it is difficult to detect a clear trend concerning this second criterion.

Table 2: Experimentation results for the city environment.

Entities	Max Group Size	CPU Gain (%)	Aggregation Cost (%)
10.000	5	38,0	5,3
10.000	10	48,0	7,1
10.000	15	54,2	8,7
10.000	20	56,0	9,0
10.000	25	54,5	8,8

The CPU gain observed in the city simulation is shown in Table 2. Like above, the CPU gain increases with the maximum size of each aggregate. This test demonstrates that the cost of the additional computations required by our approach (the *Aggregation Cost*) is limited and indeed remains much smaller than the total computation gain, even with a high number of agents. However, all the tests highlight a non linear variation of the CPU gain according to the group size. This can be explained by the actual number of agents within each aggregate during a simulation run. According to our observations, this number is generally between 10 and 15 agents, which coincides with the slowdown in growth of the CPU gain after a maximum of 15 agents per aggregate. The main explanation for this result lies in the choice of the psychological distance and the aggregation utility threshold. Because the first one is focused on the agents' short-term goal, it is sometimes too specific and greatly limits the size of the groups. The second one has been set high enough to trigger an aggregation if and only if both physical and psychological distances are low. Because of what has been said before, this induces the agents to be grouped only if they are also physically close enough. Finally, those fixed parameters lead to the small group sizes observed in our simulations.

The results of the two subway scenarios highlight an important variation of the behavioral dissimilarity between the experimentations involving a small number of entities – 100 and 300 – within the station, and those dealing with 500 to 1000 agents. Again, our observations showed that this difference is the direct result of the overcrowding of the station which becomes a key phenomenon when it contains more than 500 microscopic entities. In this situation, agents trying to pass the ticket barriers are colliding with the ones queuing at the ATMs and the ticket vending machines. The time required to access the objects is greatly increasing. Some agents even leave the station because they get upset to wait so long to use the machines or because they get stressed by the crowd. This situation does not appear during macroscopic simulations, because the aggregation itself greatly reduces the perceived density of agents in the station. Hence, our approach is not able to simulate properly specific microscopic phenomena because the aggregation process is too coarse by grouping several entities into one agents and applying to it a microscopic agent model.

5. DISCUSSION AND FUTURE WORK

In this paper, we presented a novel approach of dynamic level of detail (LOD) for large scale simulations, which breaks from the general habit of using a single level of representation. Instead, we proposed the use of behavioral LOD and we introduced the notions of dynamic change of representation and spatial aggregation. Hence, our approach can be applied to various models governing agent behavior, dealing for example with navigation, decision, or emotions. Moreover, it is able to determine by itself the most suitable representation level for each agent, regarding the simulation context.

The results detailed in section 4 show an encouraging CPU gain between the microscopic simulation and the one implementing LOD techniques, even on experimentations involving a high number of agents. Moreover, this gain leads to an acceptable behavioral dissimilarity when the number of entities within the station does not lead to crowded situations.

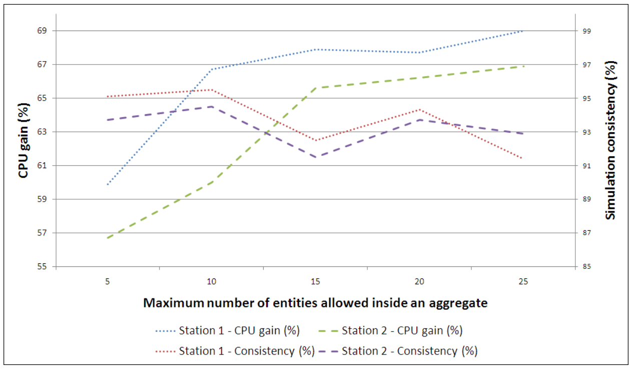

Figure 5: Evolution of CPU gain and simulation consistency for a maximum of 300 agents within each station.

However, when microscopic phenomena such as a very high density of agents are observed, the behavioral distance increases significantly. Thus, this result highlights two shortcomings of our current approach. The first one is the consequence of the assumptions made for the experimentations, where the same agent model has been used on each representation level. Doing so implies that the physical area of an aggregate to be equal to the one of an individual agent. This remark brings forward the need for a group model taking into account, at the minimum, a surface, a density and deformation factor. Of course, using a more complex model with specific group actions, knowledge and detailed internal state might help designing a more realistic simulation.

The second shortcoming of our approach results from the fact that the aggregation process, by merging several entities into a single one, may be too coarse in some situations. Although it may lead to visual inconsistencies, it can also create a strong behavioral difference between a model and another with a lower level of representation. A solution would be to use an intermediate level between several entities and an aggregate: the mesoscopic level [24]. The idea is to assume that, among the two main attributes categories defined in chapter 3.1 – physical and psychological – the first one is the most objective and observable. Thus, going from the microscopic to the mesoscopic level consists in aggregating only the psychological attributes. The mesoscopic agent will then have several bodies, corresponding to the physical microscopic bodies of the agents and driven by the low representation physical agent model (such as the navigational model), and one brain controlled by the low representation psychological model (such as decisional or emotional models). Doing so would decrease the CPU gain, because it only saves computation time on some agent models, but would also decrease the dissimilarity, in particular in crowded situations. However, many issues remain to be studied, especially the criteria for aggregating and disaggregating mesoscopic agents.

Another weakness highlighted by our experiments is the use of fixed parameters which leads to small aggregates size. This limitation could be lifted by a study on a more generic psychological distance between two agents and on the dynamicity of the most important parameters of the approach such as the aggregation and dissagregation thresholds. The first one has been arbitrarily defined and deserves to be made dependent on more subjectives parameters, such as those which are important for the user observing the simulation. For our experiments, we chose the short-term goal as criterion, but another user which may be particularly interested in the stress level of each agent might decide that two actors are psychologically close if they share the same stress level. Although this attribute cannot be used alone to define a coherent psychological distance, there is a need to give to the user some control over the weight of each psychological attribute in the computation of the distance. Secondly, the important parameters such as the thresholds were defined by hand for this first experimentation. An idea would be to set them dynamically, function of the number of representation levels, the number of agents in the simulation and the available CPU power. This way, the aggregation and desaggregation processes would adapt the context of the simulation and would provide the best CPU gain / dissimilarity ratio.

One of the most important limitations of the sub problem defined in section 3.3 is the simplification done to the scalability of the agent models part, especially for the definition of the aggregation, disaggregation and memory operators. Indeed, one of the major improvements of this work would come from the ability to obtain these operators through learning or search. As mentioned in section 3.1, the use of machine learning mechanisms can be promising. They may, for example, focus on minimizing the behavioral dissimilarity defined in section 4.

Finally, the issue of communications between agents – which relates more generally to the notion of scalability of the interactions between agents at all levels of representation – has not been directly studied in this work, as our agents do not communicate directly with each other. Considering such ability would require the definition (or the automatic search) of aggregation and disaggregation operators to transform the information emitted from an agent at a given level of representation to another at another level. If those operators are similar to the ones working on the agents' representation – except that they would work on the semantics – they would be called for each interaction and might increase the computational cost. This important point has yet to be investigated.

6. ACKNOWLEDGMENTS

We would like to thank Professor Jean-Daniel Zucker for some fruitful discussions around the notions of abstraction aggregators and meso-agents, which we intend to explore further.

7. REFERENCES

[1] D. Luebke. *A Developer's Survey of Polygonal Simplification Algorithms*. IEEE Computer Graphics and Applications. 2001.

[2] P. Langley Pat and D. Choi. *A Unified Cognitive Architecture for Physical Agents*. In proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06). 2006.

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

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

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

[6] 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.

[7] A. Treuille, S. Cooper and Z. Popovic. *Continuum crowds*. In proceedings of the 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH 2006). 2006.

[8] A. Helsinger and T. Wright. *Cougaar: A Robust Configurable Multi Agent Platform*. In proceedings of the 26th IEEE Aerospace Conference. 2005.

[9] B. Yersin, J. Maïm, J. Pettré and D. Thalmann. *Crowd Patches: Populating Large-Scale Virtual Environments for Real-Time Applications*. In proceedings of the 2009 Symposium on Interactive 3D Graphics and Games (I3D'09). 2009.

[10] I. Wright and J. Marshall. *Egocentric AI Processing for Computer Entertainment: A Real-Time Process Manager for Games*. In proceedings of the 1st International Conference on Intelligent Games and Simulation (GAME-ON 2000). 2000.

[11] J. Laird. *Extending the Soar Cognitive Architecture*. In proceedings of the 1st Conference on Artificial General Intelligence (AGI-08). 2008.

[12] P. Davis and R. Hillestad, *Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Managemen*t. In proceedings of the 25th Winter Simulation Conference (WSC'93). 1993.

[13] 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.

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

[15] R. 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.

[16] C. Brom and Z. Vlckova. *IVE: Virtual Humans' AI Prototyping Toolkit*. International Journal of Computer and Information Science and Engineering. 2007.

[17] F. Bellifemine, G. Caire, A. Poggi and G. Rimassa. *JADE: A software framework for developing multi-agent applications*. Information and Software Technology. 2008.

[18] C. Niederberger and M. Gross. *Level-of-Detail for Cognitive real-time Characters*. The Visual Computer: International Journal of Computer Graphics. 2005.

[19] S. Franklin, U. Ramamurthy, S. D'Mello, L. MacCauley, A. Negatu, R. Silva and V. Datla. *LIDA: A Computational Model of Global Workspace Theory and Developmental Learning*. In proceedings of the AAAI Fall Symposium on AI and Consciousness: Theoretical Foundations and Current Approaches. 2007.

[20] 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 Transportation Meeting (EWGT2002). 2002.

[21] 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.

[22] 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.

[23] D. Krajzewicz, M. Bonert and P. Wagner. *The Open Source Traffic Simulation Package SUMO*. In proceedings of the 10th RoboCup Infrastructure Simulation Competition (RoboCup 2006). 2006.

[24] A. Ruas. *The role of meso level for urban generalisation*. In proceedings of the 2nd Workshop of the ICA commission on Generalisation and Multiple Representation (GMR 1999). 1999.

[25] J. Maïm, B. Yersin, J. Pettré and D. Thalmann. *YaQ: An Architecture for Real-Time Navigation and Rendering of Varied Crowds*. IEEE Computer Graphics and Applications. 2009.

[26] H. Nwana, D. Ndumu and L. Lee. *ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems*. In proceedings of the 3rd International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'98). 1998