Holonic Model of a Virtual 3D Indoor Environment for Crowd Simulation

Stéphane Galland and Nicolas Gaud

Multiagent Group, IRTES-SET
Université de Technologie de Belfort-Montbéliard, France
{stephane.galland,nicolas.gaud}@utbm.fr
http://www.multiagent.fr

Abstract. Multiagent-based simulations enable us to validate different use-case scenarios in a lot of application domains. The idea is to develop a realistic virtual environment to test particular domain-specific procedures. This paper presents a holonic model — hierarchy of agents — of a physical environment for the simulation of crowds in virtual 3D buildings. The major contributions of this paper are the agentization of the environment model to support multilevel simulation, and the definition of energy-based indicators to control the execution of the model. Finally, the application of the model inside an airport terminal is presented. It permits to validate the principles of the models and the computation gains.

Keywords: Multi-agent simulation, Holonic multiagent systems, Multilevel simulation, Virtual environment, Janus platform

1 Introduction

The models for urban simulation may be classified in four main families: macroscopic, mesoscopic, microscopic and multilevel simulation models. Macroscopic simulation models are based on the deterministic relationships of the flow, speed, and density of population (peoples or traffic stream) [1]. The simulation in a macroscopic model takes place on a region-by-region basis rather than by tracking individuals. Macroscopic simulation models were originally developed to model traffic into distinct transportation networks, such as freeways, corridors (including freeways and parallel arterials), surface-street grid networks, and rural highways. This approach enables the simulation of very large population with a small relative computational cost. However, due to its high-level of representation, the results are imprecise and related to masses of population. Microscopic simulation models are concerned with the movement of people on the basis of dynamic and individual behaviors. Behaviors may be based on a large scope of models such as the intelligent driver and the lane changing models to represent the drivers, or a force-based model for the pedestrians [2–6]. These models are effective in assessing the conditions of congestion and saturation, the study of the topological configuration, and evaluating the impact of individual behavior on the system. However, these models are difficult to implement and costly in terms of computation time, and they can be difficult to calibrate. Several models use an organizational point of view, and define the system in terms of roles and organizations [7]. Mesoscopic models combine the properties of the microscopic and macroscopic models. For example, they may focus on the entities in the system by using models that do not distinguish the individuals from each other, such as particle models [8], by grouping the individuals within higher-level entities such as groups of pedestrians [9], or by using a discrete model of the environment, such as the cellular automata [10]. Multilevel models support different levels of simulation (macro, meso, micro). Different points-of-view exist on the means to integrate these different levels in a single multilevel model. Two models, one micro and one macro for instance, may be run in sequence, and the output of one is the input of the other [11]. The multilevel model may also be able to select the best simulation level dynamically, according to indicators: the more the computer has available resources, the more the selected level tends to be the micro one [12]. This paper is related to this last type of multilevel simulation.

All these models consider the agents and the environment. Nevertheless, they are often conducted to optimize the behaviors of the agents and their interactions, and using very/too simple representations of the environment. As stated by [13], the environment is an important part of the system that should be studied in details. In the rest of this paper, we focus on the *physical environment model* for the microscopic and multilevel multiagent simulation of crowds, as illustrated by Figure 1. The pedestrians' behaviors are not detailed in this paper. Two problems may occur during the execution the environment model: (i) the computational cost may be huge, and incompatible with efficient response constraints; and (ii) many times the executed algorithm is too complex and too expensive according to the topology of the crowd and obstacles; simpler and faster algorithms could be used in place with an equivalent answer quality. Several models and platforms were proposed to solve these problems and answer to the complex problem of the simulation of crowds in real-time: GAMA [14], Breve [15], FLAME... According to our knowledge, none of them is providing a model of the environment with a holistic, nor the multilevel modeling we want to apply.

In this paper, an agent-oriented model of the environment is defined. Note that in the rest of this paper, the term "agent" refers to the agents, which are supporting the environment model; in opposition to the "application agents," which represent the pedestrians. Why is an agent-oriented model used for the environment? It permits to adapt the overall environment's behavior dynamically, during its execution. *The use of agents enables to evaluate and to predict the computational costs of the algorithms locally, and to select the one, which is fitting the constraints in time and in quality.* A specific type of agent is considered: the holon[1] [16, 7]. Why are the holons used for the environment model?

[1] Holon: an agent composed of agents, which can be seen as an atomic entity from its outside, and an entity composed by sub-holons from its inside, at the same time.

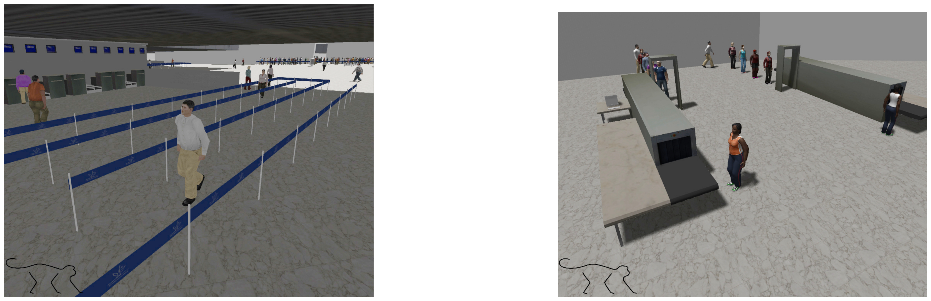

Fig. 1. Screenshots of the Airport simulation, provided by the Simulate® commercial tool

They enable to support the dynamics, and the intrinsic hierarchical nature of the physical environment. This agent-oriented model is qualified of holonic, and defined according to the CRIO metamodel and the associated holonic framework[2] [7].

This paper is structured as follows: Section 2 presents the organizational model of the environment. Section 3 describes the agents/holons that are supporting the environment model. Section 4 presents the energy-based indicators that are involved in the multilevel simulation. Section 5 describes the application of our environment model on the simulation of an airport. Section 6 concludes this paper.

2 Organizational Model of the Environment

Works on urban or 3D environments are numerous and mainly used for performances. They require highly specialized calculations and therefore, a significant number of resources. As shown in [17], the environment is often distributed according to places. A place is a semi-closed spatial area bounded by static objects (usually walls). Each place may have connections called portals, with its neighbor places. They are used to ease the interaction between two adjacent spaces. They also permit to use structural environment models such as Potentially Visible Set [17] for improving the computation of the perceptions of the application agents. Places are basically defined a priori by the designer of the simulation. They generally correspond to the structural decomposition of the environment [18, 19]. Entities are objects inside the environment, and are located in a single place through a dedicated data-structure (usually a spatial tree or a spatial grid).

To simulate large and complex worlds, it is important to support unbalanced places in terms of entities they are containing. Indeed, the difference of population coverage by the places may cause fewer global performances to the simulator.

[2] The CRIO metamodel and the holonic framework are outside the scope of this paper. See `http://www.aspecs.org` or [7] for details.

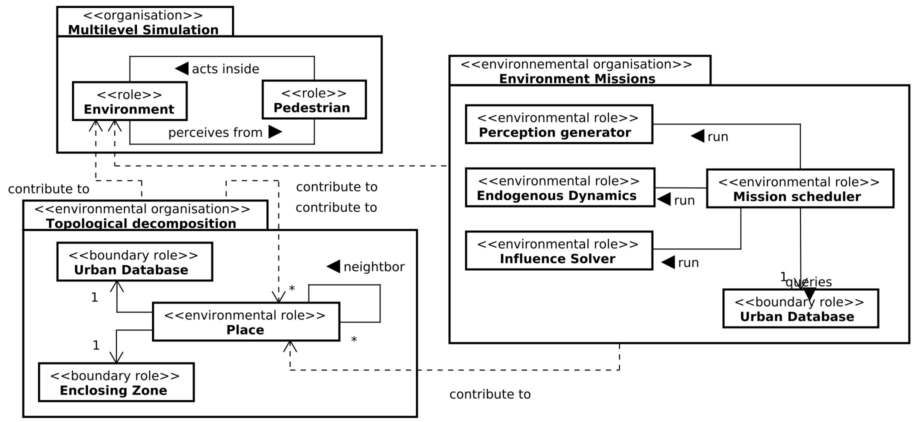

Fig. 2. Organizations and roles of the environment, using the formalism defined in [7], and on `aspecs.org`

To overcome this problem, places are decomposed in turn into a collection of dynamically built zones. In contrast to the statically defined decomposition, these zones are built during the simulation process.

Figure 2 shows the organizations that compose the environment system. In a global point of view, the `Multilevel Simulation` organization defines the overall simulation system. Two roles are defined inside. The `Pedestrian` role is played by the agents who are participating in the simulation, *i.e.* the pedestrians. The `Environment` role is played by any agent or group of agents responsible for the overall behavior of the physical environment. Interactions between them are based on the influence-reaction model [20]; and on the computation of the pedestrian's perceptions [6]. Each player of the `Environment` role must have the capacity [7] to compute perceptions to each pedestrian. The `Environment`'s players must also have the capacity to gather influences — wishes of actions — from each pedestrian.

The `Topological decomposition` organization focuses on the structure of the physical environment itself. This organization provides the capacities required by the `Environment` role in the previous organization. The `Topological decomposition` organization can contribute to the behavior of this higher-level role. The `Topological decomposition` organization is composed of interconnected places. Each of them is responsible for the environment's missions [13] in the considered space. It also manages the objects inside the zone. To realize its behavior, a `Place` role must interact with the role `Urban Database` to obtain and to change the information related to the objects inside the environment.

The role `Enclosing zone` supports the multilevel modeling of the environment. The organization `Topological decomposition` represents a level within the hierarchy of composition of the environment. It is necessary that each level in this hierarchy has access to information dedicated to the multilevel dynamics. As a *boundary role*, the role `Enclosing zone` is responsible for providing to a place the state of the enclosing zone, and the indicators and the constraints given by this higher level. All these information will be described in more detail later in this paper.

The organization `Environment Mission`, inspired by [13], defines all the roles required to satisfy all the missions of the environment for a specific place. An instance of this organization is integrated as a group in all the agents, which are playing the role `Place`. This link between the two organizations is represented by the relationship "`contribute to`" in Figure 2.

The next step is to identify the agents and their behaviors in order to obtain the agents' society, which exhibits the expecting behavior of the organizations, and the roles described above.

3 Agents of the Environment

Figure 3 illustrates an instance of a society of agents, who may execute the environment behavior. The key point is to determine, for each role, if a standalone agent or a group of agents[3] is playing it. When one agent is managing an entire place, it is playing the role `Place` in the `Environment Model`. When a place needs to be split and managed by a group of agents, one of them must play the role `Place` in the `Topological decomposition`, and `Mission scheduler` in the `Environment Mission` organizations. The decision to decompose or not a place is the responsibility to the agent playing the role `Place`. It depends on: (i) the individual indicators, which are specific to an agent playing the role `Place`; and (ii) the indicators shared in the context of a group of agents, which is an instance of the organization `Topological decomposition`. Each agent playing the role `Place` can access to these indicators by interacting with the role `Enclosing zone`. These indicators are detailed in Section 4.

Figure 4 illustrates the state machine of the agents of the environment. This state machine describes the composition-decomposition behavior of the agents. Events `isCollapsable` and `isDecomposable` correspond to the detection of a change from the agent situation according to the indicators described in the next section. They correspond respectively to the events of composition and decomposition.

When an agent H decides to decompose the place z associated to it, it applies the algorithm for create sub-holons that are managing the different sub-zones of z. A group `topological decomposition` is created and populated by agents playing the role `Place`, one for each sub-zone. The function `updateGlobal-IndicatorsForSubAgents` updates the indicators that are used by the sub-agents for their own decomposition decisions.

When an agent H decides that the place z should not be split, it destroys its sub-holons. The group `Topological decomposition` is destroyed. A group

[3] Note that a holon may represent either an atomic agent or a composed agent [16].

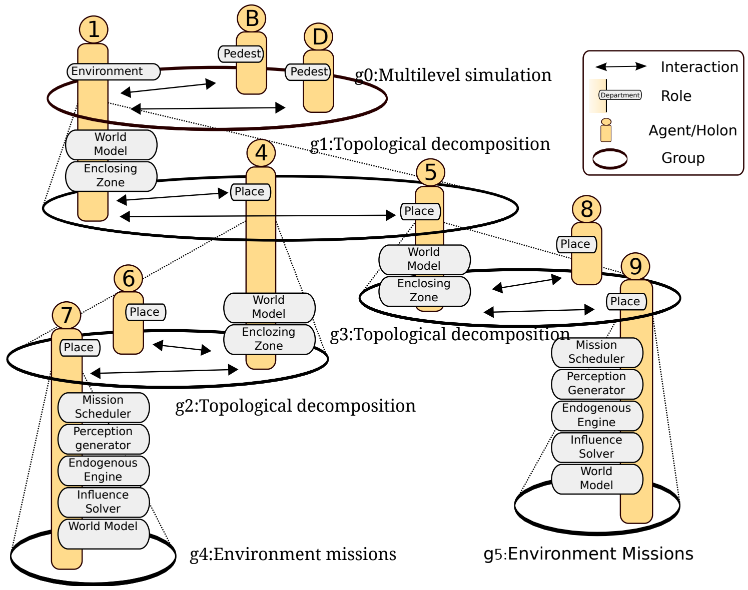

Fig. 3. Example of agent society, which is managing the environment.

associated to the organization `Environment Missions` is created to allow the super-agent to reach its main goals: determining the perceptions of the agents, and managing the influences from them.

Both algorithms can build, level by level and during the run-time, the hierarchical model of the environment. The evaluation of the indicators is performed continuously during the simulation process. The holarchy[4] of the environment may change dynamically while being influenced by the movement of the pedestrians, and by the resources available for the simulation.

4 Indicators for the Multilevel Simulation

In this paper, we propose three main families of indicators for evaluating the events `isCollapsable` and `isDecomposable`:

– The **mass of a zone** indicates the importance of a place of the environment for the simulation. This value depends on the scenario. For example, it may be proportional to the density of pedestrians in the place, or depends upon the presence of an immersed human user in this place.
– The **structural depth** describes the minimum or the maximum depth of the decomposition of a zone. Thus, it is possible for a role `Place` to restrict the depth of its topological decomposition.

[4] Holarchy: a hierarchy of holons that may intersect other holarchies by sharing holons together.

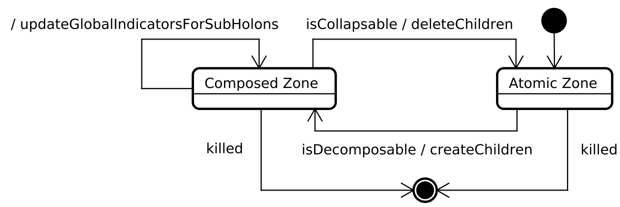

Fig. 4. State machine for the hierarchical behavior of each agent of the environment

- The **resource constraint** describes the limits of the available resources for a place to achieve its simulation. This constraint allows considering low-level information, close to the operating system, such as the computation time. It is possible to impose a time constraint for approaching a real-time execution. A resource constraint can also describe the limits for any type of low-level resource (memory, network bandwidth…)

The mass of the object e describes the importance of e at an instant of the simulation. More massive an object is, the more it influences the simulation results, and it consumes resources. This mass, denoted M_e is defined by Equation 1, where w_e is the constant mass of e.

$$M_e = \begin{cases} w_e & \text{if } e \text{ is an atomic object} \\ \sum_{a \in e} w_a & \text{if } e \text{ is a composed object} \end{cases} \tag{1}$$

The mass of a zone z describes the importance of z during the simulation. It is defined by Equation 2. More massive a place is, the more it is involved, and it influences the results for the simulation. The mass of z is proportional to the mass of the sub-places and the objects therein.

$$M_z = \alpha_z.w_z + \sum_{a \in D_z} \alpha_a.M_a + \sum_{e \in E_z} \alpha_e.w_e \tag{2}$$

D_z is the set of sub-places of z. E_z contains the objects located on z. w_z is the constant mass of z, given by the designer of the simulation model. It represents the importance of the place in the scenario. w_e is the constant mass of the object environment e. α_i is the weight of i (z, a and e) when it contributes to M_z. The set of weights is constrained by $\left(\sum_{i \in \{z\} \cup D_z \cup E_z} \alpha_i\right) = 1$.

The resource constraint R_α is imposed by the super-agent to its sub-agents. It represents the amount of available resources for the sub-agents. Its computation is based upon the use of a weight-based function, and is depending upon the mass of the sub-places. The resource constraint for a sub-agent a of the agent z is defined by:

$$R_a = (R_z - k_z) \times \frac{M_a}{\sum_{b \in D_z} M_b} \qquad \forall a \in D_z \tag{3}$$

k_z is a constant, which estimates the consumption of resources by the super-agent to run its decision-making algorithms.

4.1 Dynamics of the Environment Agents

At every instant of the simulation, the environment agents evaluate the indicators described above. This evaluation determines if they should change of state: being a manager of a decomposed place, or the manager of an atomic place.

As shown in the state machine in Figure 4, each agent is facing with one of the following decisions:

Case 1 If the agent manages an atomic place, must it decompose this place and create sub-agents?

Case 2 If the agent manages a decomposed place, must it combine the sub-places, and destroy the sub-agents managing these sub-places?

In case 1, the agent can be decomposed if there are enough resources to the execution of its sub-agents. Equation 4 describes the condition triggering the change of state of the agent (`isDecomposable` becomes true). A super-agent must decompose when it has sufficient resources at its disposal, or the evaluation of the consistency between simulations at the levels n and $n+1$ indicates that the super-agent does not approximate correctly any more the behaviors of its sub-agents.

$$\begin{aligned}\left[\left(\exists a \in D_z, \left|Eg_z - Eg_a\right| > \epsilon\right) \vee \left(\forall R, R_z \geq \sum_{p \in D_z} \mathrm{g}_R(p) + k_z\right)\right] \wedge \\ \left(\max_z < i \vee \min_z > i_z\right) \wedge \\ \left(E_z \neq \emptyset\right)\end{aligned} \tag{4}$$

The first member of the equation permits to evaluate the consistency of the simulation. The energies of the sub-agents are computed and compared with the energy of the super-agent. If the difference between the energies of two levels exceeds the constant error ϵ, then the super-agent does not more approximate accurately its sub-agents. The energy terms Eg_z and Eg_a are application-dependent, and are illustrated later. The second member of this equation is based on the use of the function $\mathrm{g}_R : D_z \rightarrow \mathbb{R}$ for estimating the amount of resources needed for executing the missions of the environment in the sub-agent p. This function g_R is dependent upon the target application. Each super-agent consumes resources for computing the various multilevel indicators, and applying the decomposition policy. This amount of resources consumed is given by the constant k_z. The constants $\min_z$ and $\max_z$ represent the minimum and maximum depths in the hierarchical decomposition of the environment.

In case 2, the agent is decomposed into a set of sub-agents managing the sub-places of z, the place associated with the super-agent. This determines whether to retain its sub-agents or destroy them. This last case corresponds to a change of the state of the super-agent. A super-agent can destroy its members when it does not have enough resources at its disposal to carry out the simulation and the evaluation of the consistency between the simulations at the levels n and $n+1$.

$$\left(\forall a \in D_z, \left|Eg_z - Eg_a\right| \leq \epsilon\right) \wedge \left(\forall R, R_z < \sum_{p \in D_z} \mathrm{g}_R(p) + k_z\right) \wedge \min{}_z < i \quad (5)$$

If the simulation has all the required resources, it is done at the most accurate level. In other words, the agents of the level n (the deepest level in the holarchy) are always executed. However, if resources become insufficient, the simulator can identify the places that require a priority allocation of the available resources. The indicators in each super-agent are used to identify which sub-agent's behaviors are too much approximated.

5 Experiments

This section describes several experiments with the agent-based environment model on the simulation of airport halls (illustrated by Figure 1). The purpose of these experiments is to ensure that our model provides similar results than other airport simulation models [21], and to evaluate the impact of the approximation applied by the use of energy indicators. The airport terminal is composed of two halls (around 0.25km^2), which are separated by gates. Each of these gates is a check point between the public place and the boarding place. The pedestrians behave according to the force-based algorithm proposed by [22].

The behavior of the agents is decomposed on three majors activities: (i) going to check-in desk, 2/3 of the passengers need to check in the baggage, and 1/3 have only hand-baggage; (ii) passing the check points; and (iii) boarding. Figure 5 illustrates the evolution of the number of passengers at the check points, and the average waiting time of these passengers. The first peak corresponds to the passengers that are not going at the check-in desks. The second/higher peak corresponds to the passengers that were at the check-in desks.

The energy at the different levels is computed according to a low-level resource criterion. When the resource criterion is at 100%, it means that the computer has enough resources to run the simulation at the finest level. When the resource is down at 60%, it means that 60% percent of the simulation may be run at the micro/finest level, and 40% percent of the simulation is run/approximate at a higher level. As explained in the previous section, the energy evaluation depends on the application. Equation 6 details a simple evaluation of this energy for the airport application. Intuitively, this energy assesses the quality of generated perceptions by the environment: more objects are not included in the perception, compared with the most accurate perception; less is the quality of the perception. $p^{\odot}$ is the set of perceived objects that are found when it is computed at the lowest level. $p^{\ominus}$ (resp. $p^{\oplus}$) represents the objects that are lost (resp. added) at a higher level in the holarchy. α_{po} and β_{po} are calibration variables. Our experiments shows that $\alpha_{po} = 1$ and $\beta_{po} = \frac{1}{|E|}$, where $|E|$ is the total number of entities in the airport, may be used by default.

$$Eg_\alpha = \begin{cases} \dfrac{\alpha_{po}|p^\ominus| + \beta_{po}|p^\oplus|}{|p^\odot|} & \text{if } p^\odot \neq \emptyset \\ \alpha_{po}|p^\ominus| + \beta_{po}|p^\oplus| & \text{else} \end{cases} \tag{6}$$

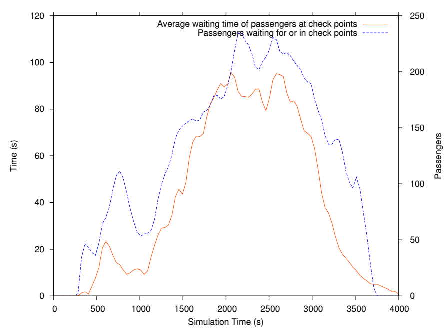

Fig. 5. Average waiting time and passenger density at the check points

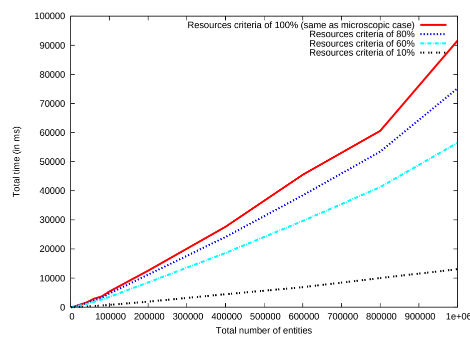

Fig. 6. Running Time according to the Environment's Levels (available resources)

The tests are performed with a set of 2,000 entities in the entry hall and 1,000 entities in the boarding place. Four check points are assumed to be available. The average computation time for one simulation step of the object-oriented model of JaSim (the original one) is 25.9 seconds, the equivalent agent-oriented model (proposed in this paper) takes 41.5 seconds with a single place for the entire place, and 8.1 seconds with two places. Figure 6 illustrates the running time of the agent-oriented model when the computational resources are limited. When this resource criterion is at 100%, it means that the computer has enough resources to run the simulation at the finer level. When the resource is down at 60%, it means that only 60% percent of the micro-simulation may be run at the finer level.

6 Conclusion

Multiagent-based simulations enable us to validate different use-case scenarios in a lot of application domains. The idea is to develop a realistic virtual environment to test particular domain-specific procedures.

This paper presents an agent-oriented and multilevel model of a situated environment for the simulation of a crowd in a virtual $3D$ building. The major contributions in this paper are, in one hand, an agent-oriented model of the physical environment, based on the holarchy, and on the other hand, a collection of energy-based indicators for evaluating the accuracy of the multilevel simulation. The model is successfully applied to the simulation of two airport halls.

These experiments permit to evaluate the impact of the multilevel simulation on the simulation results, and the gain in terms of computational costs.

The energy formula presented within this paper may be generalized to become application-independent. One possible direction is to provide formula for classes of environments, which may be used to build applications. We consider that the energy indicators may be interesting to distribute the agents other a computer network also. In this paper, we propose to use energy-based indicators. Other types of indicators may be used in place to obtain accurate evaluations: $\mathbb{Z}$ function... Finally, the proposed model may be applied on large-scale systems to evaluate the approximation introduced by our multilevel model. Our model must also be compared to existing multiagent simulation frameworks (GAMA, MatSIM, FLAME...).

Acknowledgments

The airport screenshots were produced, in conjunction with the platform Janus[5], by the commercial tool Simulate of the Voxelia SAS[6] company, France. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Voxelia SAS.

References

1. Helbing, D., Treiber, M.: Gas-Kinetic-Based Traffic Model Explaining Observed Hysteretic Phase Transition. Phys. Rev. Lett. **81**(14) (Oct 1998) 3042–3045
2. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM (1987) 25–34
3. Thalmann, D., Musse, S.R.: Crowd simulation. Springer (2007)
4. Dey, P., Roberts, D.: A Conceptual Framework for Modelling Crowd Behaviour. In: DS-RT '07: Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications, Washington, DC, USA, IEEE Computer Society (2007) 193–200
5. Razavi, S.N., Gaud, N., Mozayani, N., Koukam, A.: Multi-agent based simulations using Fast Multipole Method: Application to large scale simulations of flocking dynamical systems. Artificial Intelligence Review **35**(1) (jan 2011) 53–72
6. Galland, S., Gaud, N., Demange, J., Koukam, A.: Environment model for multiagent-based simulation of 3D urban systems. In: the 7th European Workshop on Multiagent Systems (EUMAS09), Ayia Napa, Cyprus (dec 2009) Paper 36.
7. Cossentino, M., Gaud, N., Hilaire, V., Galland, S., Koukam, A.: ASPECS: an agent-oriented software process for engineering complex systems - how to design agent societies under a holonic perspective. Autonomous Agents and Multi-Agent Systems **2**(2) (mar 2010) 260–304

[5] `http://www.janus-project.org`
[6] `http://www.voxelia.com`

8. Schaefer, L.A., Mackulak, G.T., Cochran, J., Cherilla, J.L.: Application of a general particle system model to movement of pedestrians and vehicles. In: WSC '98: Proceedings of the 30th conference on Winter simulation, Los Alamitos, CA, USA, IEEE Computer Society Press (1998) 1155–1160
9. Gaud, N., Galland, S., Gechter, F., Hilaire, V., Koukam, A.: Holonic multilevel simulation of complex systems: Application to real-time pedestrians simulation in virtual urban environment. Simulation Modelling Practice and Theory **16**(10) (November 2008) 1659–1676
10. Karafyllidis, I., Thanailakis, A.: A model for predicting forest fire spreading using cellular automata. Ecological Modelling **99**(1) (1997) 87 – 97
11. Burghout, W., Koutsopoulos, H.N., Andréasson, I.: Hybrid Mesoscopic-Microscopic Traffic Simulation. Transportation Research Record: Journal of the Transportation Research Board **1934** (2005) 218–255
12. Gaud, N., Galland, S., Koukam, A.: Towards a Multilevel Simulation Approach Based on Holonic Multiagent Systems. In: UKSIM '08: Proceedings of the Tenth International Conference on Computer Modeling and Simulation. (2008) 180–185
13. Weyns, D., Parunak, H.V.D., Michel, F., Holvoet, T., Ferber, J.: Environments for Multiagent Systems State-of-the-Art and Research Challenges. In: Third International Workshop (E4MAS 2006). Volume 4389 of Lecture Notes in Artificial Intelligence. Springer, Hakodate, Japan (may 2006) 1–47
14. Grignard, A., Taillandier, P., Gaudou, B., Vo, D., Huynh, N., Drogoul, A.: GAMA 1.6: Advancing the art of complex agent-based modeling and simulation. In Boella, G., Elkind, E., Savarimuthu, B., Dignum, F., Purvis, M., eds.: PRIMA 2013: Principles and Practice of Multi-Agent Systems. Volume 8291 of LNCS. Springer Berlin Heidelberg (2013) 117–131
15. Klein, J.: breve: a 3D simulation environment for the simulation of decentralized systems and artificial life. In: 8th International Conference on the Simulation and Synthesis of Living Systems, MIT Press (2002)
16. Rodriguez, S.A.: From analysis to design of Holonic Multi-Agent Systems: a Framework, methodological guidelines and applications. PhD thesis, Université de Technologie de Belfort-Montbéliard (2005)
17. Cohen-Or, D., Chrysanthou, Y.L., Silva, C.T., Durand, F.: A survey of visibility for walkthrough applications. IEEE Transactions on Visualization and Computer Graphics **9**(3) (2003) 412–431
18. Farenc, N.: An informed environment for inhabited city simulation. PhD thesis, Lausanne (2001)
19. Vanbergue, D.: Conception de simulation multi-agents: application à la simulation des migrations intra-urbaines de la ville de Bogota. PhD thesis, Université Paris VI (December 2003)
20. Michel, F.: The IRM4S model: the influence/reaction principle for multiagent based simulation. In: Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07), ACM (May 2007)
21. Jackou, R.C.: Contribution à la Gestion des Opérations de la Sûreté Aéroportuaire : Modélisation et Optimisation. Phd thesis, Institut National Polytechnique de Toulouse, Toulouse, France (novembre 2010)
22. Buisson, J., Galland, S., Gaud, N., Gonçalves, M., Koukam, A.: Real-time collision avoidance for pedestrian and bicyclist simulation: a smooth and predictive approach. In: 2nd International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTRANS13), Halifax, Nova Scotia, Canada, Elsevier (June 2013) 815–820 Best Paper Award.