Organizational and Holonic Modelling of a Simulated and Synthetic Spatial Environment

Stéphane Galland and Nicolas Gaud

Univ. Bourgogne Franche-Comté, UTBM, IRTES EA7274, F-90010 Belfort, 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 simulated 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 corresponding computational gains.

Keywords: Multi-agent simulation, Holonic multiagent systems, Multilevel simulation, Simulated Environment, Physic Environment, Virtual environment, SARL language

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) [23]. 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 not very accurate and strongly related to masses of population. Microscopic simulation models are concerned with the movement of people on the basis of dynamic 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 [36, 46, 10, 35, 17]. 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 term of computation time, and they can be difficult to calibrate. 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 [44], by grouping the individuals within higher-level entities such as groups of pedestrians [19], or by using a discrete model of the simulated environment, such as the cellular automata [25]. Multilevel models support different levels of simulation (macro, meso, micro). Different points-of-view exist on the means to integrate these different levels into 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 [5]. The multilevel model may also be able to select the best simulation level dynamically, according to specific indicators: the more the computer has available resources, the more the selected level tends to be the micro one [20]. This paper is related to this last type of multilevel simulation.

In this paper, authors focus on multiagent-based simulation, and more specifically on the modeling of the simulated environment. As stated by [51], the environment is an important part of a multiagent system that should be studied in details. In the rest of this paper, we focus on the *simulated physical environment* for the microscopic and multilevel multiagent-based simulation of crowds, as illustrated by Figure 1. The pedestrians' behaviors are not detailed in this paper. Two main common problems may occur during the execution of the simulated environment model: (i) the computational cost may be too huge, and thus 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 accuracy. Several models and platforms were proposed to address these issues: GAMA [22], Breve [26], FLAME, etc. According to our knowledge, none of them is providing a holistic multilevel model of the simulated environment including the hierarchical spatial decomposition, the management of the transitions between levels, the interrelationships among the environmental components and their associated dynamics.

This paper introduces an agent-oriented multilevel model of the environment for a multiagent-based simulation. 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 (in our airport simulation, or the vehicles in traffic simulation). Why is an agent-oriented model used for the simulated environment? It permits to adapt the overall simulated 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] [41, 8].

[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.

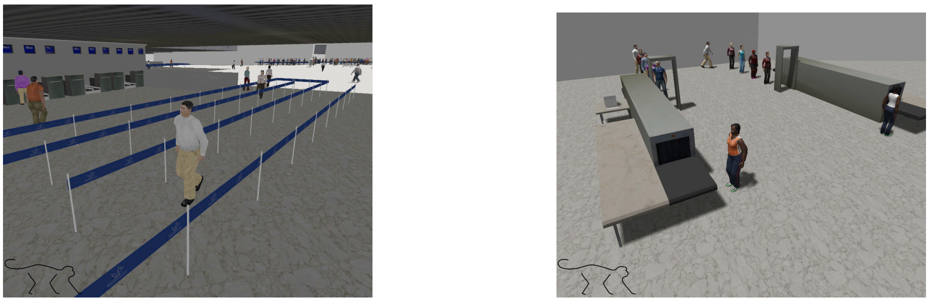


Fig. 1. Screenshots of the Airport simulation, provided by the Simulate® commercial tool.

Why are the holons used for the simulated environment model? They enable to support the intrinsic hierarchical nature of the simulated physical environment and its dynamics. This agent-oriented model is qualified of holonic, and defined according to the CRIO metamodel and the associated holonic framework[2] [8]. CRIO is an organizational metamodel enabling the hierarchical modeling of a system in terms of organizations and roles.

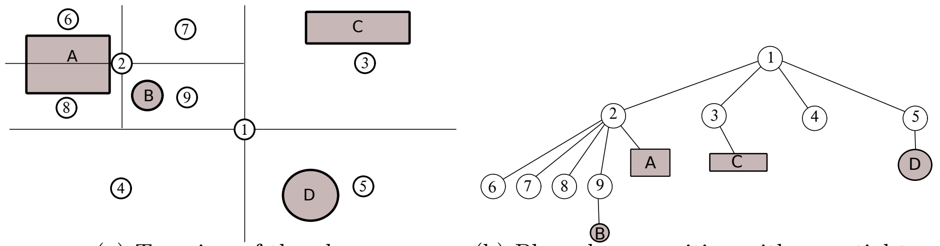
This paper is structured as follows: Section 2 presents the organizational model of the simulated environment. Section 3 describes the agents/holons that are supporting the environment model. Section 4 presents the energy-based indicators that are involved in the multilevel simulation. Section 5 introduces a basic implementation of the holonic environment model using the SARL programming language. Section 6 describes the application of our environment model on the simulation of an airport. Section 7 relates the content of this paper to existing works. Section 8 links our works to the identified challenges related to the environment. And finally, Section 9 concludes this paper, and provides perspectives of this work and new challenges.

2 Organizational Model of the Simulated Environment

Urban systems are typical examples of complex systems. Urban simulations quickly require important computational resources if the user want to maintain a high level of accuracy. As shown in [7], the simulated environment is often distributed into a collections of places to easily distribute computational costs. 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 simulated environment models such as Potentially Visible Set [7] for improving the computation of the visual perceptions of the application agents.

[2] The CRIO metamodel and the holonic framework are outside the scope of this paper. See `http://www.aspecs.org` or [8] for details.

Places are basically defined a priori by the designer of the simulation. They generally correspond to the structural decomposition of the simulated environment with connected graphs [12, 48]. Entities are objects inside the simulated environment, and are located in a single place through a dedicated data-structure (usually a spatial tree or a spatial grid, see Figure 2).



(a) Top view of the place (b) Place decomposition with a spatial tree

Fig. 2. Decomposition of a place in areas with a spatial tree

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 lower global performances to the simulator. 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 according to the population density in each zone.

Figure 3 shows the various organizations that compose the proposed environment model. In a global point of view, the `Multilevel Simulation` organization defines the overall simulation system according to two main roles. The `Pedestrian` role is played by the agents, which are participating in the simulation, *i.e.* the pedestrians, and describes their corresponding behaviors. The `Environment` role is played by any agent or group of agents that is responsible for the overall behavior of the simulated physical environment. Interactions between them are based on the influence-reaction model [29]; and on the computation of the pedestrian's perceptions [17]. Each player of the `Environment` role must have the capacity [8] to compute the perceptions for each pedestrian usually, by exploring a spatial data-structure (see Figure 2). 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 simulated physical environment itself. This organization provides the capacities that are 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

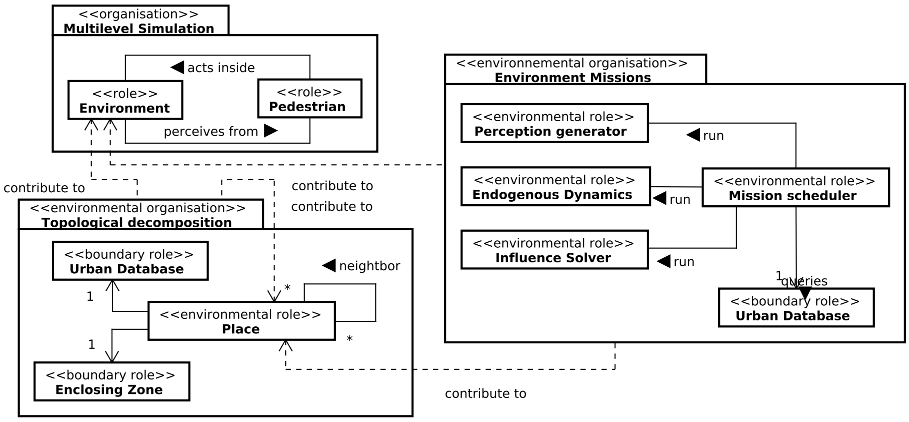


Fig. 3. Organizations and roles of the simulated environment, using the formalism defined in [8], and on `aspecs.org`

missions [51] in the considered geographical 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 update the information related to the objects inside the simulated environment.

The role `Enclosing zone` supports the multilevel modeling of the simulated environment. The organization `Topological decomposition` represents a level within the hierarchy of composition of the simulated 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 considered zone, as well as transferring the indicators and the constraints given by the upper level to the various places, which compose it, if any. All these information will be detailed later in this paper.

The organization `Environment Mission`, inspired by [51], defines the roles that are required to satisfy all the missions of the simulated environment for a specific place. An instance of this organization is supported as a group in the agents, which are playing the role `Place`. This link between the two organizations is represented by the relationship "`contribute to`" in Figure 3.

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

3 Agents of the Simulated Environment

Figure 4 illustrates an instance of a society of agents, which may execute the simulated environment behavior. The key point is to determine, for each role,

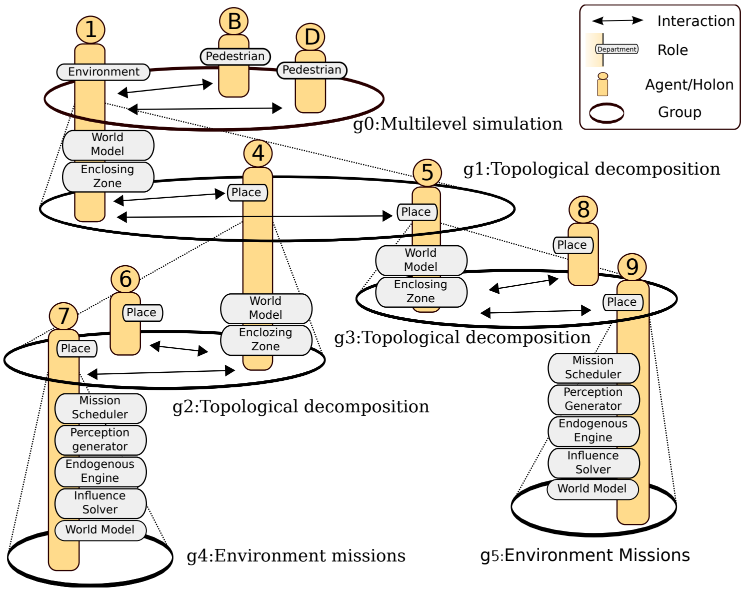


Fig. 4. Example of an agent society, which manages the simulated environment.

if a standalone agent or a group of agents[3] is playing it. When a single 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 of 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 5 illustrates the state machine of the agents of the simulated environment. This state machine describes the composition (resp. 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 creating sub-holons that are managing the different sub-zones of z (see Algorithm 1). A group `topological decomposition` is created and populated by agents playing the role `Place`, one for each sub-zone. The function

[3] Note that a holon may represent either an atomic agent or a composed agent [41].

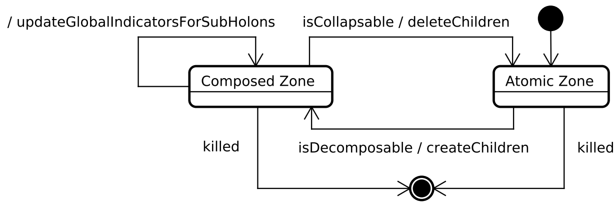


Fig. 5. State machine for the hierarchical behavior of each agent of the simulated environment

`updateGlobalIndicatorsForSubAgents` updates the indicators that are used by the sub-agents for their own decomposition decisions.

```
g := getGroup(H, "Environment missions")
if g ≠ nil then
    releaseRole(H, "Mission scheduler", g)
    releaseRole(H, "Perception generator", g)
    releaseRole(H, "Endogenous Dynamics", g)
    releaseRole(H, "Influence Solver", g)
    releaseRole(H, "Urban Database", g)
end if

S := computeSubZonesOf(z)
g := createGroup("Topological decomposition")
requestRole(H, "Urban Database", g)
requestRole(H, "Enclosing Zone", g)
updateGlobalIndicatorsForSubAgents(H,S)
foreach zone in S do
    agent := createAgentIn(H)
    requestRole(agent, "Place", g, zone)
done
E_z := ∅
```

Algorithm 1: Algorithm for the decomposition of an agent associated to a zone of the simulated environment.

When an agent H decides that the place z should not be split, it destroys its sub-holons (see Algorithm 2). The group `Topological decomposition` is destroyed. A group associated to the organization `Environment Missions` is then created to enable the super-agent to reach its main two 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 simulated environment. The evaluation of the indicators

```
g := getGroup(H, "Topological decomposition")
if g ≠ nil then
    foreach agent in getPlayers(g,"Place") do
        kill(agent)
    done
    releaseRole(H, "Enclosing Zone", g)
    releaseRole(H, "Urban Database", g)
    E_z := createSuperBodies()
end if

g := createGroup("Environment Missions")
requestRole(H, "Mission Scheduler", g)
requestRole(H, "Perception Generator", g)
requestRole(H, "Endogenous Dynamics", g)
requestRole(H, "Influence Solver", g)
requestRole(H, "Urban Database", g)
```

Algorithm 2: Algorithm for the composition of an agent associated to a zone of the simulated environment.

is performed continuously during the simulation process. The holarchy[4] of the simulated environment may change dynamically, while being influenced by the movements of the pedestrians, and by the resources (machine memory, processing resource, etc.) that are available for the simulation.

The proposed algorithms assume that the super-agent group, which is participing in the organization `Environment Missions`, has all the necessary capacities required by this organization. Each of these capabilities is the realization of one of the missions of the simulated environment. In other words, the super-agent offers a service (a capacity according to the CRIO metamodel) to simulate these missions. An alternative is the definition of sub-agents dedicated to the support of each mission of the simulated environment. The super-agent always plays the role `Mission scheduler`. However, it delegates the other missions to its sub-agents. Thus, each mission could be implemented natively by a service in each agent, or by a whole group of interacting agents. This last possibility is not detailed in this paper.

4 Indicators for the Multilevel Simulation

In this paper, authors propose three main classes of indicators for triggering the events `isCollapsable` and `isDecomposable`:

– The **mass of a zone** indicates the relative importance of a place of the simulated environment for the whole simulation. This value obviously depends on the considered scenario. For example, it may be proportional to

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

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.
- 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, etc.)

The mass of the object e describes the importance of e at a given 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.

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

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 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 $\sum\limits_{i \in \{z\} \cup D_z \cup E_z} \alpha_i = 1$.

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:

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

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 Simulated Environment Agents

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

As shown in the state machine presented in Figure 5, 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 for the execution of its sub-agents. Equation 4 describes the condition that triggers 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 the behaviors of its sub-agents, any more.

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

The first member of the equation evaluates 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 threshold ϵ, then the super-agent's behavior does not approximate accurately its sub-agents' behaviors. The energy terms Eg_z and Eg_a are application-dependent, and are illustrated later in this paper. The second member of Equation 4 is based on the use of the function $\mathrm{g}_R : D_z \to \mathbb{R}$, for estimating the amount of resources that are needed for executing the simulated environment missions by the sub-agent p. This function g_R depends upon the target application. Each super-agent consumes resources for computing the various multilevel indicators, and applying the decomposition policy. This amount of consumed resources 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 simulated 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 for carrying out the simulation,

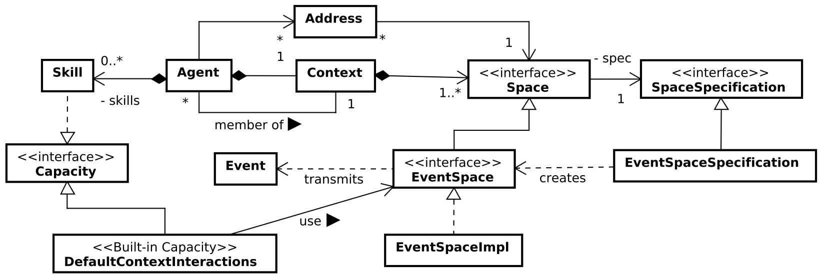


Fig. 6. Major concepts in the SARL metamodel.

and the evaluation of the consistency between the simulation results 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 for identifying the sub-agent's behaviors that are too approximated.

5 Implementation with sarl

In this section, the implementation with the SARL programming language of the proposed holonic environment model is discussed.

SARL[5] is a general-purpose agent-oriented programming language [40]. This language aims at providing the fundamental abstracts for dealing with concurrency, distribution, interaction, decentralization, reactivity, autonomy and dynamic reconfiguration. These high-level features are now considered as the major requirements for an easy and practical implementation of modern complex software applications. The main perspective that guided the creation of SARL is the establishment of an open and easily extensible language. Such language should thus provide a reduced set of key concepts that focuses solely on the principles considered as essential to implement a multi-agent system. In this paper, two elements of the metamodel of SARL are extensively used: Agent and Space. These two concepts are illustrated on Figure 6.

In order to take into account heterogeneous interaction models, SARL proposes the `Space` concept, which is an *interaction space*. Space is the abstract

[5] `http://www.sarl.io`

to define *an interaction space between agents or between agents and their environment*. This concept is used for defining the interaction between an agent and the physical simulated environment. In the SARL toolkit, a concrete default space, which propagates events, called `EventSpace` (and its implementation `EventSpaceImpl`), is proposed.

An agent is an autonomous entity having a set of skills to realize the capacities it exhibits. An agent has a set of built-in capacities considered essential to respect the commonly accepted competences of agents, such autonomy, reactivity, proactivity and social capacities. Among these capacities, the agent can incorporate a collection of behaviors that will determine its global conduct. An agent has also a default behavior directly described within its definition. *A Behavior maps a collection of perceptions represented by Events to a sequence of Actions.* In the default configuration of SARL, the various behaviors of an agent communicate using an event-driven approach. *An Event is the specification of some occurrence in a Space that may potentially trigger effects by a listener* (e.g. agent, behavior, etc.)

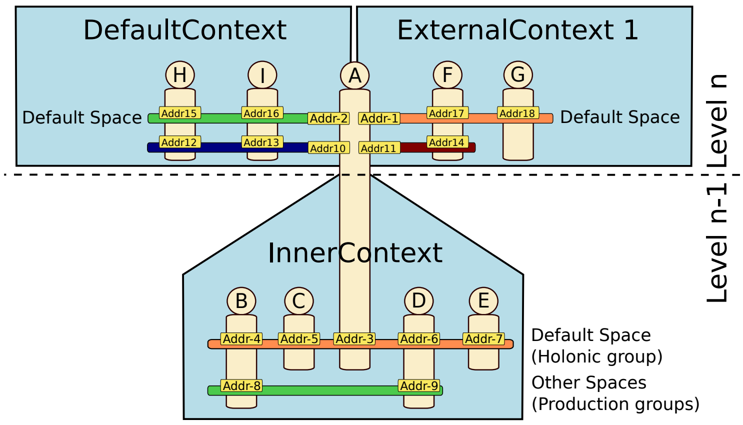


Fig. 7. A holon or a recursive agent in SARL.

In SARL, agents can be composed of other agents. Therefore, SARL agents are in fact holons that can compose each other to define hierarchical or recursive MAS, called holarchies. In order to achieve this, SARL agents are self-similar structures that compose each other via their Contexts. Each agent defines its own Context, called *Inner Context* and it is part of one or more *External Contexts*. For instance in Figure 7, agent A is taking part of two *External Contexts* (i.e. Default Context and External Context 1) and has its own *Inner Context* where agents B, C, D, and E evolve.

6 Experiments

This section describes several experiments with the proposed simulated environment model in the simulation of an airport halls (illustrated by Figure 1). The airport terminal is composed of two halls, which are separated by gates. Each of these gates is a check point between the public area and the boarding area.

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 their baggages, and 1/3 have only hand-baggages; (ii) passing the check points; and (iii) boarding. Figure 8 illustrates the evolution of the number of passengers at the check points, and the average waiting time of these passengers. The first peaks correspond to the passengers that are not going at the check-in desks. The second/highers peaks corresponds to the passengers that were at the check-in desks. Figure 9 shows the evaluation of the energy according to different levels of available computational resources. When this resource critera 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 only 60% percent of the micro-simulation may be run at the finest 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 the generated perceptions by the simulated environment: more objects are not included in the perception, compared with the more precised possible perception, less is the quality of the perception. $p^{\odot}$ is the set of the 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 $\alpha_{po} = 1$ et $\beta_{po} = \frac{1}{|E|}$, where $|E|$ is the total number of entities in the airport, may be used by default.

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

The tests are performed with a set of 2,000 entities in the entry hall and 1,000 entities in the boarding area. 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 [17]) is 25.9 seconds, the equivalent agent-oriented model (proposed in this paper) takes 41.5 seconds with a single place for the entire area, and 8.1 seconds with two places. Figure 10 illustrates a comparison of the running time of the agent-oriented model and the original object-oriented model. The two curves have a similar shape, due to the use of the same low-level algorithms (perception and action computations) on the same types of data-structures (quadtree). The agent-oriented approach provides a curve nearest to the linear curve than the original approach. This is mainly due to a better balancing of the nodes of the two trees, one for each place, than the balancing of the single tree of the original model.

Figure 11 illustrates the running time of the agent-oriented model when the computational resources are limited. When this resource critera is at 100%, it

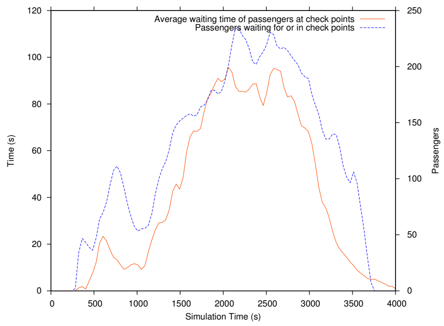


Fig. 8. Avarage waiting time and passenger density at the check points.

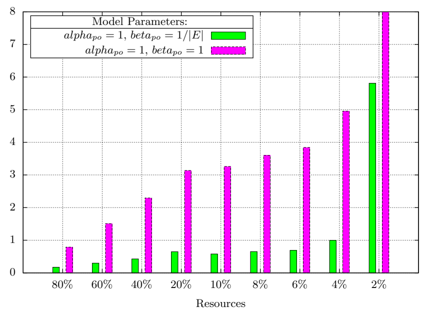


Fig. 9. Evaluation of the Average Energy

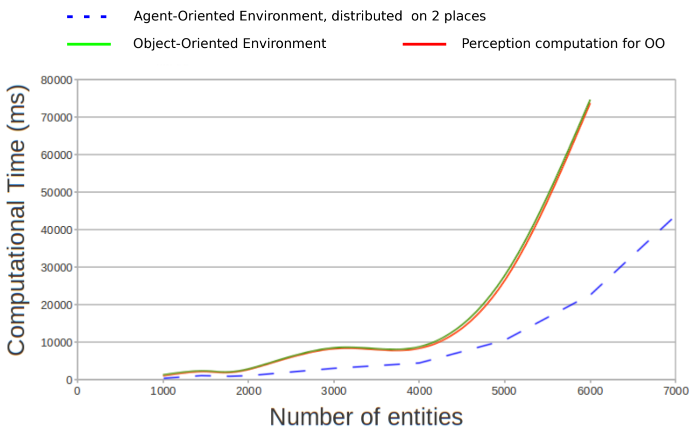


Fig. 10. Running Time of the Object-Oriented and the Agent-Oriented Models.

means that the computer has enough resources to run the simulation at a 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.

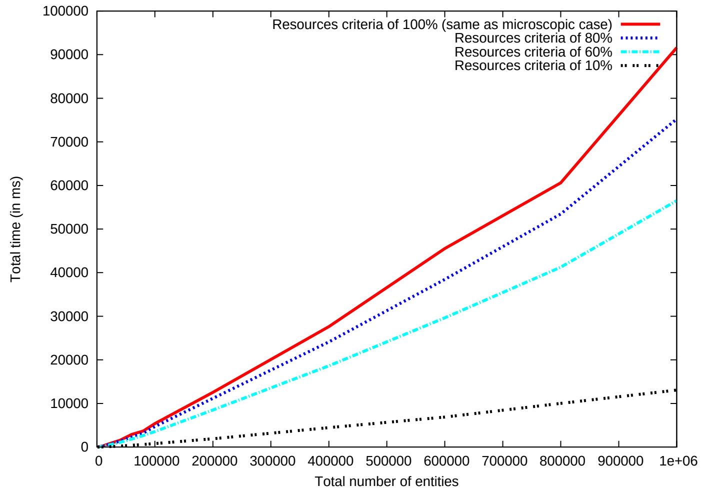


Fig. 11. Running Time according to the Simulated Environment's Levels (available resources).

7 Related Works

Our inspirations for the simulated physical environment are the models for the simulation of crowds and traffic into virtual environments [31, 11, 45, 2, 18]. The Artifact [37], CArtAgO [38] and smart object [46] models are also an inspiration. They propose similar interaction models between agents and objects in the environment, and the definition of the latter.

A taxonomy of virtual environments is provided by [6]. A virtual environment is synthetic and simulated. It may be classified according to three dimensions. The first dimension is the type of environment: access directly to real environments, or virtual environment. The second dimension characterizes the nature of the agents interacting in the system environment: Human, physical agents (robots), or virtual agents (animats, virtual characters). The third dimension describes the models of interactions that are used by the agents. A wide range of models may be considered from stigmergy [32] to social and organizational interactions. Your model is designed for representing a simulated physic environment. It provides the mechanisms for computing the perception and managing the actions of the agents according to physic laws (gravity, collision avoidance, etc.) Our model is related to the domains "multiagent-based simulation" that is defined by [6]. In our application (including our holonic environment model), the environment and the agents are simulated. The interaction is based on the stigmergy principles. Additionally, our environment model is related to the "interactive simulation" domain [6]. Indeed, the body concept that is used in our model enables to replace any virtual agent associated with a body by a Human, who is controlling this body with the same interaction principles: the body that is controlled by the Human must emit influences [29] as the ones controlled by the virtual agents.

The problems related to the interaction between an agent, and the physical environment have been treated with different perspectives. One of the models used in our approach is the Influence-Reaction model [29]. It supports the simultaneity of the actions in an environment by considering the interactions initiated by agents as uncertain, and detecting and resolving the conflicts between the interactions. This approach can be compared to the artifact concept [37], which proposes to model the objects in the environment. They provide a set of actions that can be applied on each of them. A similar model named smart object is proposed for virtual environments [46]. In your approach, the influences related to spatial traveling, and those dedicated to trigger actions are distinct for enabling a detailed specification of the parameters for each of them. The IODA model [27] and its extension PADAWAN [28] allow modeling the interactions between the agents and the environment by assuming that every entity in the environment is an agent. Our model is partially incompatible with this vision in the context of the physical environment modeling. Indeed, the bodies of the agents are not agents.

Several organizational approaches consider the environment [14, 21, 33]. In the context of this paper, the key element is the modeling of the environment with a holarchy. The space concept that is provided by the SARL programming language serves as an abstract for organizational groups and spatial areas.

8 Links to Research Challenges related to the Agent Environment

Since 10 years, research challenges are identified for the environment in multiagent systems. This section provides several links and possible answers to a couple of these challenges.

The first research challenge concerns the **difference between the agent environment and the agents that inhabit it** [51]. Based on the principle that agents are autonomous entities, and the environment does not contain autonomous entities, the question of defining what is an agent and what is not arisen. The model presented in this paper is based on the principle of separation between the mind and the body. This distinction in the context of artificial intelligence was mainly proposed in robotics [3]. Its application to multiagent systems, where there is not necessarily a physical body, has not been examined in detail in the literature [42]. However, this concept has been used occasionally to model and constrain the interactions between the agents and the environment

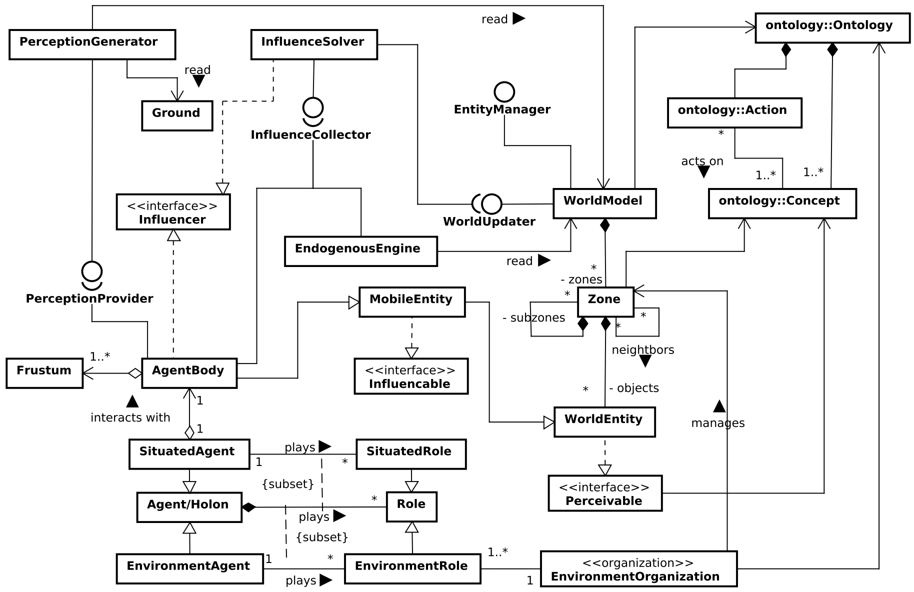


Fig. 12. Concepts for the definition of an agent environment, and the relationship between the agents and this environment [17, 2, 9]

[29, 9, 2, 43, 1, 39, 47]. In these works, the body is an object of the environment, with a dynamic that cannot be controlled directly by the agent. The agent environment controls the dynamics' properties of the body (position, orientation, etc.), and ensures that these properties follow the rules and laws of the Universe [30, 34]. Nevertheless, every agent is able influencing its body using a mechanism such as the Influence-Reaction model [29]. Consequently, in our model, *every entity that physically exists is an object of the environment, including the bodies of the agents. Agents become the autonomous entities that control the bodies.* They are able to perceive and acts in the agent environment through their bodies. The major mode of interaction that is considered in our model is the stigmergy.

The second research challenge is related to the **taxonomy of the agent environments** [51]. In previous works, we have considered different dimensions of the agent environment: *the physic dimension, communication dimension, and social/organizational dimension* [16, 15]. The first dimension concerns the environments that are represented the physical world. The second dimension contains the communication tools and infrastructures. The third dimension defines the social and organizational relationship between the agents. This paper treats only of the agent environments that are related to the first category. Moreover, since the agent environment is a model of the real world, it is simulated and synthetic.

The third challenge concerns **the definition of the abstracts and concepts that may be used for defining an agent environment** [49, 24]. The model presented in this paper is based on a meta-model that provides abstracts for defining the topology of the agent environment, and the objects that are located in this environment [17, 2, 9]. Figure 12 provides an overview of the concepts that are at the heart of the holonic model presented in this paper. The physical agent environment is intrinsically hierarchical [13]. The zone concept is proposed for supporting the decomposition of this environment into interconnected sub-zones, that could be decomposed in turn. The objects (`WorldEntity`) are located in a zone of the environment. All these objects could be perceived by the agents (`Perceivable`). And, several of them could have their state changed by agent actions (`Influencable`), according to the Influence-Reaction model [29]. The agent body is an object that could be controlled by an agent (`SituatedAgent`), contains a field-of-view (`Frustum`) that could be used for computing the perception, and provides the functions for emitting influences in the agent environment. In the context of this paper, the dynamics of the agent environment are managed by specific agents (`EnvironmentAgent`), including the internal processes related to the physical agent environment (`EndogenousEngine`). These agents are responsible for supporting the missions of the environments defined in [50, 51]. They are members of a holarchy (see Section 2) for supporting the hierarchical nature of the physical environment.

The **handling of the interferences between the agents' actions** is a research challenge [24]. In the context of a simulated agent environment, *the Influence-Reaction model [29] provides the framework for detecting and solving the conflicts among the influences – uncertain actions – given by the agents.* Unfortunately, the Influence-Reaction model does not give a detailed model for detecting and solving these conflicts. In our agent environment model, the laws of the Universe are known, and correspond to the Newtow laws. Consequently, the *influences are forces when they describe a motion, and action triggers for executing specific actions on the objects,* e.g. pushing a button. *A physic engine is used for computing the reactions of the agent environment related to the motion of the objects* [4]. This approach enables our agent environment to preserve its integrity, which is one of its responsibilities [51].

The second responsibility of the environment is **ensuring the locally of the perception and the actions** [51]. *The perception mechanism is based on a field-of-view* (named `Frustum` in Figure 12). The shape of the field-of-view is defined by geometrical elements that have a position relative to the position of the agent's body in the physical environment. This definition ensures a local perception for the agents. In our model, *the actions are local since they are always related to an object of the physical agent environment: move the object, do an action on the object.*

The environment includes a broad diversity of logical functionalities. The **3-layer model** was proposed for structuring them [51]. The first layer is dedicated to the MAS application. It is composed by the agents, the agent environment, and the MAS framework. The second layer contains the middleware and the operating system. The third layer is related to the infrastructure definition (hosts, network infrastructure, etc.) The works presented in this paper focus of the first

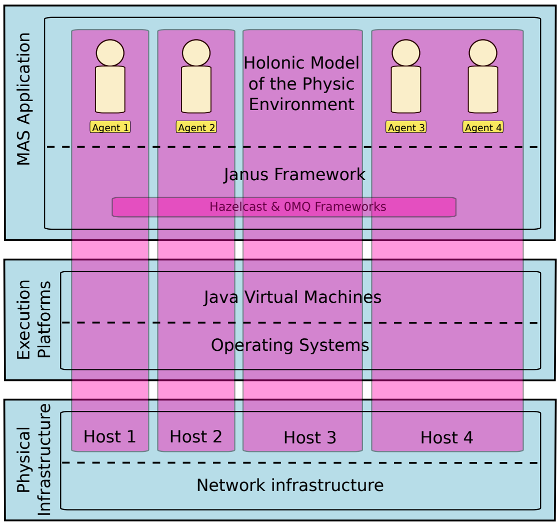


Fig. 13. The 3-layer model with the holonic environment model, and the JANUS framework

layer, as it is illustrated by Figure 13. Indeed, a model of the agent environment is proposed. This model is implemented in the SARL language[6]. And, the resulting MAS runs on the JANUS framework[7]. This framework supports the distribution of the MAS application over a computer network by using the Hazelcast and 0MQ libraries. According to the specifications of the SARL language, the agents are naturally distributed, and are designed accordingly. The agents are not aware of the means that are used for implementing the distribution by the MAS framework.

The last challenge is related to the **need of a specific language for describing the agent environment** [24]. In the context of this paper, no answer is given to this challenge. However, our experiences in the modeling of the dimensions of the environments with the SARL language give a partial answer [16, 15]. It is possible to describe the dynamics of the physic and communication dimensions of the agent environment. Unfortunately, it is still difficult to describe the topology of this environment. *We consider that a specific language is still needed for describing the agent environment.* This language may be based on the works done for Artifact [37] and CArtAgO [38], for instance.

[6] `http://www.sarl.io`
[7] `http://www.janusproject.io`

9 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/simulated environment to test particular domain-specific procedures. This paper presents an agent-oriented and multilevel model of a situated simulated 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 simulated 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 cost.

The energy formula presented within this paper may be generalized to become application-independent. One possible direction is to provide formula for classes of simulated 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, etc).

Our holonic model of the physic agent environment provides answers to specific research challenges that are identified during the past 10 years (see Section 8). Concepts that are used for describing the objects, and the topology of the physical agent environment are proposed: zone, environmental object, body, etc. The use of the CRIO organizational approach enables to define the behaviors related to the environment modules for supporting the agent environment responsibilities. The dynamics of this environment are supported by holons, which also mimic the intrinsic hierarchical nature of the environment. In our opinion, two major challenges are still under research activities: How to handle large-scale systems with a physical environment in multiagent-based simulations? What is the language for defining the elements, the topology, and the dynamics of the agent environment that could be a simulated environment or the Reality?

Acknowledgments

The airport screenshots were produced, in conjunction with the platform JANUS[8], by the commercial tool SIMULATE of the VOXELIA SAS[9] 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.

[8] `http://www.janusproject.io`
[9] `http://www.voxelia.com`

References

1. Barella, A., Ricci, A., Boissier, O., Carrascosa, C.: MAM5: Multi-agent model for intelligent virtual environments. In: 10th European Workshop on Multi-Agent Systems (EUMAS 2012). pp. 16–30 (2012)
2. Behe, F., Galland, S., Gaud, N., Nicolle, C., Koukam, A.: An ontology-based metamodel for multiagent-based simulations. International Journal on Simulation Modelling, Practice, and Theory 40, 64–85 (Jan 2014), http://authors.elsevier.com/sd/article/S1569190X13001342
3. Brooks, R.A.: Intelligence without representation. Artificial intelligence 47(1), 139–159 (1991)
4. Buisson, J.: An environment model for the multi-agent simulation of mobility in urban areas. Phd thesis, Université de Technologie de Belfort-Montbéliard, Belfort, France (december 2014), in French
5. Burghout, W., Koutsopoulos, H.N., Andréasson, I.: Hybrid Mesoscopic-Microscopic Traffic Simulation. Transportation Research Record: Journal of the Transportation Research Board 1934, 218–255 (2005)
6. Carrascosa, C., Klügl, F., Ricci, A.: Virtual environments 4 mas. In: International Workshop on Environments for Multiagent Systems (E4MAS14). Springer (May 2014)
7. 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), 412–431 (2003)
8. 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), 260–304 (mar 2010)
9. Demange, J., Galland, S., Koukam, A.: Analysis and design of multi-level virtual indoor environment. Int. Journal Systemics and Informatics World Network 10, 145–152 (Jul 2010), http://siwn.org.uk/
10. 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. pp. 193–200. IEEE Computer Society, Washington, DC, USA (2007)
11. Donikian, S., Paris, S.: Towards embodied and situated virtual humans. Motion in Games, Lecture Notes in Computer Science 5277, 51–62 (2008)
12. Farenc, N.: An informed environment for inhabited city simulation. Ph.D. thesis, Lausanne (2001)
13. Farenc, N., Boulic, R., Thalmann, D.: An informed environment dedicated to the simulation of virtual humans in urban context. In: Proceedings of EUROGRAPHICS 99. pp. 309–318 (1999)
14. Ferber, J., Michel, F., Baez, J.: Agre: Integrating environments with organizations. In: Environments for Multi-Agent Systems, Lecture Notes in Computer Science, vol. 3374, pp. 48–56. Springer Berlin Heidelberg (2005), http://dx.doi.org/10.1007/978-3-540-32259-7_2
15. Galland, S., Balbo, F., Gaud, N., Rodriguez, S., Picard, G., Boissier, O.: Contextualize agent interactions by combining social and physical dimensions in the environment. In: Demazeau, Y., Decker, K. (eds.) 13th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS) (Jun 2015)

16. Galland, S., Balbo, F., Gaud, N., Rodriguez, S., Picard, G., Boissier, O.: A multidimensional environment implementation for enhancing agent interaction. In: Bordini, R., Elkind, E. (eds.) Autonomous Agents and Multiagent Systems (AAMAS15). Istanbul, Turkey (May 2015)
17. 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
18. Galland, S., Knapen, L., Yasar, A.u.h., Gaud, N., Janssens, D., Lamotte, O., Koukam, A., Wets, G.: Multi-agent simulation of individual mobility behavior in carpooling. International Journal on Transport Research Part C 45, 83–98 (Aug 2014), `http://www.sciencedirect.com/science/article/pii/S0968090X14000035`
19. 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), 1659–1676 (Nov 2008)
20. 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. pp. 180–185 (2008)
21. Gouaïch, A., Michel, F.: Towards a unified view of the environment(s) within multiagent systems. Informatica 29(4), 423–432 (may 2005)
22. 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, LNCS, vol. 8291, pp. 117–131. Springer Berlin Heidelberg (2013)
23. Helbing, D., Treiber, M.: Gas-Kinetic-Based Traffic Model Explaining Observed Hysteretic Phase Transition. Phys. Rev. Lett. 81(14), 3042–3045 (Oct 1998)
24. Helleboogh, A., Vizzari, G., Uhrmacher, A., Michel, F.: Modeling dynamic environments in multiagent simulation. Int. Journal on Autonomous Agents and Multiagent Systems 14(1) (2007)
25. Karafyllidis, I., Thanailakis, A.: A model for predicting forest fire spreading using cellular automata. Ecological Modelling 99(1), 87 – 97 (1997)
26. 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)
27. Kubera, Y., Mathieu, P., Picault, S.: Ioda: an interaction-oriented approach for multi-agent based simulations. Autonomous Agents and Multi-Agent Systems 23(3), 303–343 (2011), `http://dx.doi.org/10.1007/s10458-010-9164-z`
28. Maudet, A., Touya, G., Duchêne, C., Picault, S.: Representation of interactions in a multi-level multi-agent model for cartography constraint solving. In: Demazeau, Y., Zambonelli, F., Corchado, J., Bajo, J. (eds.) Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection, Lecture Notes in Computer Science, vol. 8473, pp. 183–194. Springer International Publishing (2014), `http://dx.doi.org/10.1007/978-3-319-07551-8_16`
29. 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)

30. Okuyama, F., Bordini, R., Rocha Costa, A.C.: ELMS: An environment description language for multi-agent simulations. In: Weyns, D., van Dyke Parunak, H., Michel, F. (eds.) First International Workshop on Environments for Multiagent Systems (E4MAS). pp. 67–83. No. 3374 in Lecture Notes In Artificial Intelligence, Springer-Verlag, Berlin, Germany (Jul 2005)
31. Paris, S., Donikian, S., Bonvalet, N.: Environmental abstraction and path planning techniques for realistic crowd simulation. Computer Animation and Virtual Worlds 17(3–4) (2006)
32. Parunak, H.: Making swarming happen. In: Conf. on Swarming and Network Enabled Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR). McLean, Virginia, USA (Jan 2003)
33. Piunti, M., Ricci, A., Boissier, O., Hübner, J.: Embodying organisations in multi-agent work environments. In: IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology (WI-IAT 2009). Milan, Italy. (2009)
34. Platon, E., Sabouret, N., Honiden, S.: Environmental support for tag interactions. In: Weyns, D., van Dyke Parunak, H., Michel, F. (eds.) Third International Workshop on Environments for Multiagent Systems (E4MAS). pp. 106–123. No. 4389 in Lecture Notes in Computer Science, Springer-Verlag, Hakodate, Japan (2006)
35. 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), 53–72 (jan 2011)
36. 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. pp. 25–34. ACM, New York, NY, USA (1987)
37. Ricci, A., Viroli, M., Omicini, A.: Programming MAS with artifacts. In: International Workshop on Programming Multi-Agent Systems. Springer Verlag (July 2005)
38. Ricci, A., Viroli, M., Omicini, A.: CArtAgO: A framework for prototyping artifact-based environments in MAS. In: E4MAS. Springer Verlag (May 2007)
39. Rincon, J., Garcia, E., Julian, V., Carrascosa, C.: Developing Adaptive Agents Situated in Intelligent Virtual Environments. In: International Conference on Hybrid Artificial Intelligence Systems. pp. 98–109. Springer (2014)
40. Rodriguez, S., Gaud, N., Galland, S.: SARL: a general-purpose agent-oriented programming language. In: International Work on Intelligent Agent Technology (IAT) (2014), to be published
41. Rodriguez, S.A.: From analysis to design of Holonic Multi-Agent Systems: a Framework, methodological guidelines and applications. Ph.D. thesis, Université de Technologie de Belfort-Montbéliard (2005)
42. Saunier, J.: Bridging the gap between agent and environment: the missing body. In: International Workshop on Environments for Multiagent Systems (E4MAS14). IFAAMAS, Springer, Paris, France (May 2014)
43. Saunier, J., Jones, H.: Mixed agent/social dynamics for emotion computation. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. pp. 645–652. International Foundation for Autonomous Agents and Multiagent Systems (2014)
44. 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. pp. 1155–1160. IEEE Computer Society Press, Los Alamitos, CA, USA (1998)

45. Tamminga, G., Knoppers, P., Lint, van, H.: Open traffic: a toolbox for traffic research. In: 3nd International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTRANS14). Springer (Jun 2014)
46. Thalmann, D., Musse, S.R.: Crowd simulation. Springer (2007)
47. Thiebaux, M., Marsella, S., Marshall, A., Kallmann, M.: Smartbody: Behavior realization for embodied conversational agents. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1. pp. 151–158 (2008)
48. Vanbergue, D.: Conception de simulation multi-agents: application à la simulation des migrations intra-urbaines de la ville de Bogota. Ph.D. thesis, Université Paris VI (December 2003)
49. Viroli, M., Holvoet, T., Ricci, A., Schelfthout, K., Zambonelli, F.: Infrastructures for the environment of multiagent system. Int. Journal on Autonomous Agents and Multiagent Systems 14(1) (2007)
50. Weyns, D., Ominici, A., Odell, J.: Environment as a first-class abstraction in multiagent systems. Int. Journal on Autonomous Agents and Multiagent Systems 14(1), 5–30 (2007)
51. 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), Lecture Notes in Artificial Intelligence, vol. 4389, pp. 1–47. Springer, Hakodate, Japan (may 2006)