# Toward an Environment for the Simulation of Heterogeneous Entities in Virtual Cities

Jocelyn Buisson[12], Nicolas Gaud[1], Stéphane Galland[1], Mikaël Gonçalves[2], and Abderrafiaa Koukam[1]

[1]Multiagent Group, IRTES-SET
Université de Technologie de Belfort Montbéliard, France
http://www.multiagent.fr

[2]Voxelia S.A.S., Belfort, France
http://www.voxelia.com

**Abstract.** The study of the individual mobility phenomena is significant in the fields of the development of urban sites, the study of security, the architecture, and the flow analysis. Whatever the studied environment, indoor or outdoor, the multi-agent approach is a highly suitable tool to study the dynamics of displacement. This paper presents an environment model, which is dedicated to the simulation in 2D or 3D virtual environments. It is primarily designed for the simulation of crowds and traffic. The model combines the agent-oriented approach with algorithms used in the fields of virtual reality simulation and video games. Our model is designed to provide the mechanisms of realistic perceptions and actions to agents. It is a software framework that provides a collection of tools to ease the implementation, and make faster the simulation of heterogeneous entities (pedestrians, cyclists, drivers...) in a virtual world.

**Keywords:** Virtual environment, Multi-agent simulation, Heterogenous entities, Urban Application

## 1 Introduction

The study of the individual mobility phenomena attracts significant interest in the development of urban sites, the study of security, the architecture, and the flow analysis. Two dominant classes of applications are the evacuation simulation, and the flow analysis of a large number of individuals in indoor or outdoor environments [1]. The purpose of these simulations is to help policy makers and experts to understand the "relationship between the organization of space and the human behaviors" [2]. The considered environments are heterogeneous: streets [3, 4], buildings [5, 6], subways [7], boats [8], airplanes [9], stadiums [10], and airports [11]. In simulations of crowds and traffic, environments are usually studied in two main views. The first concerns the evolution of the structure of the environment [12, 13], and the second deals with the social activities, including traffic analysis and movement of crowds [14, 15]. The study of these views

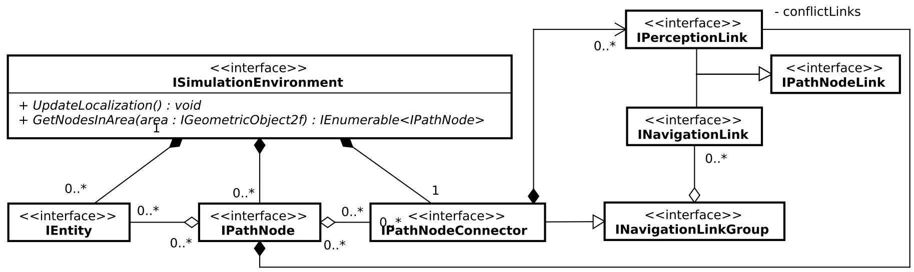


**Fig. 1.** General class diagram for the environment.

requires to support not only the macroscopic aspects, but also the entities at a microscopic level. Whatever the studied environment, indoor or outdoor, the multi-agent approach is a highly suitable tool to study the dynamics of displacement. Multi-agent systems are more flexible than macroscopic models, based on differential equations, to simulate spatial and evolutionary phenomena [16].

The environment is widely considered as one of the essential parts of a multi-agent simulation. However, there are different perspectives on the role it plays, and on its definition. The environments used in our works can be considered as special cases of "physical environments" [17, 18]. The notion of "physical environment" refers to the class of systems in which the agents and the objects have an explicit position, and the perceptions and actions also localized [19].

This paper presents an application with a concrete implementation of a physical environment that is dedicated to the simulation in 2D or 3D virtual environments. It is considered inaccessible, non deterministic, dynamic and continuous [20]. It is primarily designed for the simulation of crowds and traffic. The model combines the agent-oriented approach with object-oriented structures, and the algorithms conventionally used in the fields of virtual reality simulation and video games. Our model is designed to provide the mechanisms of realistic perceptions and actions to agents. It is a software framework that provides a collection of tools to ease the implementation, and to make faster the simulation of heterogeneous entities (pedestrians, cyclists, drivers. . . ) in a virtual world. The missions of the environment in a multi-agent simulation led to the proposal of an abstract architecture for the model of this environment. The proposed architecture is conceptually similar to that proposed by [18] while focusing on the modeling of virtual worlds. Our goal is to minimize the approximation problem, which is introduced by the modeling and the simulation [21] in the context of virtual environments.

Several models and platforms are already proposed for the simulation of heterogeneous entities in urban environments [22]. The novelty of our approach is in the highly detailed model of the environment, which may be used at the same time by the simulator, i.e. the agents, and the interactive 3D rendering tools. For example, VISIM and AIMSUN provide powerful simulation environments;

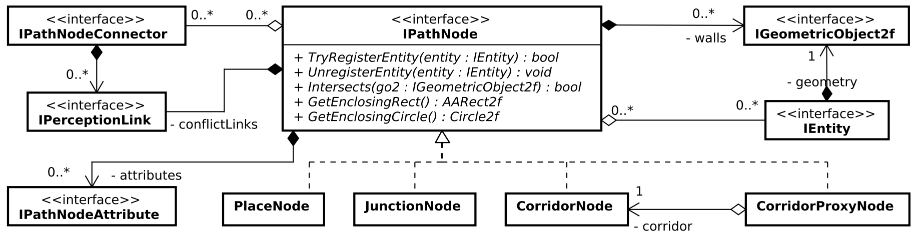


**Fig. 2.** Class diagram for the displacement surfaces.

nevertheless, they are still difficult to support the physical behavior of the components of a vehicle (wheels, springs. . . ), and the driver behavior, in real-time.

This paper is structured as follows: Section 2 presents the structural definition of the environment model. Section 3 details the dynamics of the environment. Experiments are briefly presented in Section 4. Section 5 concludes this paper and provides perspectives.

## 2 Structure of the Environment

The environment is modeled as a hyper-graph containing two layers. Figure 1 shows the corresponding class diagram. The first layer provides a view on the navigation. The second layer is dedicated to the perception. The structure of the environment is composed of three main elements: the nodes (`IPathNode`), which model the displacement *surfaces* (Section 2.1); the *edges* (`IPathNodeLink`), modeling the links between the surfaces (Section 2.2); and the *entities* (`IEntity`), which are the mobile and immobile objects within the environment. The environment can be modeled in 3D or 2D.

Our environment model fulfills its mission of sharing information with all of its nodes and its links. The nodes, in which all entities are located (including non-physical entities, stored to provide information to agents), and the links can be used to guide the agents toward their goals. The environment also allows the sharing of information between agents through publicly exposed attributes that can be attached to their body (the entity that is attached to the agent) [23].

### 2.1 Displacement surfaces

The nodes are regions of space in which agents can *move freely*. These regions are bounded either by walls, *i.e.* elements that must be considered as permanent obstacles (even though they may not represent a physical barrier), or connectors, which allow the navigation to the adjacent nodes. The nodes can have any geometrical shape (convex or concave). It is preferable to choose convex shapes for easier navigation of the entities, particularly pedestrians. Indeed, the movement algorithms are sensitive to the topology of the environment [24]. Each node

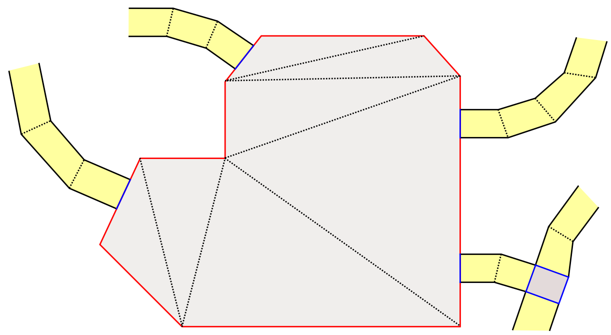

**Fig. 3.** A place connected to four corridors. Connectors are shown in blue, the walls in red.

stores references to all the entities, such as the bodies of agents or the static obstacles, that have their bounding volume intersecting the surface of the node. Consequently, an entity may be referenced by more than one node at a time. These references are automatically updated by the environment at the beginning of each simulation step. Even if a node can represent any surface, three main types of nodes are proposed and used: corridor, place, junction. Their geometries are sufficiently simple for real-time applications, and significantly accurate to represent most of the indoor or outdoor environments.

**Corridor** The corridor is a node constructed from the extrusion of a cubic Bezier curve, which is then discreted into quadrilateral sections, as illustrated on Figure 3. The corridor is defined by two control points at each of its ends. These control points, in addition to positions, contain the tangent vectors of the curves before and after the control points. The distance of extrusion is interpolated along the node thereby represent enlargements or narrowing of the surfaces, according to these tangents' attributes. A corridor is attached to, at most, four connectors. Two opposite sides are connected with simple connectors for enabling the navigation and the perception between adjacent zones. The two other opposite sides may be attached to lane-changing connectors if the node represents a road lane. The corridor is usually the type of node that is the most used, since it allows to represent the roads, sidewalks and even the rails. In addition, because the primary primitive is a curve, it is possible to generate a network of sidewalks, roads or rails automatically from the vector data of a geographic information system.

**Place** The place is a "generic" node that represents a generally large area with any number of inputs and outputs. Its geometric representation is any polygon (concave or convex, but without crossing or hole), illustrated by Figure 3. Any number of connectors can be attached to the border of the place, under the condition that none of them are intersecting. This place is mainly used to represent

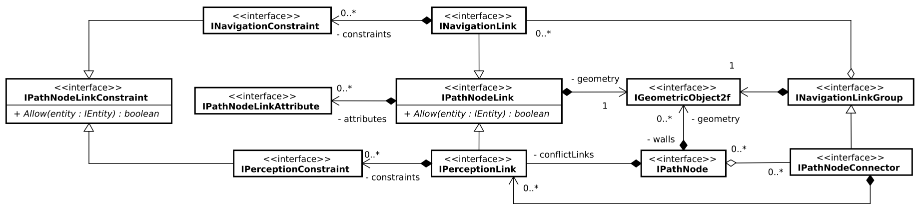


**Fig. 4.** Class diagram for the definition of the connectors.

open spaces, or large internal parts of a building which cannot be well modeled using corridors only.

**Junction** The junction node is a small quadrilateral shape with simple connectors as borders. The reason for the existence of this type of node is simply to help the creation of the connections between the corridors. It allows the creation of crossings of type "L," "T" or "X." Figure 3 illustrates a "T" junction. This type of node is a key element to represent the pedestrian crossings in urban environments, or to join the corridors inside a building.

## 2.2 Connections between the displacement zones

The links in the graph are modeled as geometric objects, which are representing the connection between two nodes. This is mostly composed of a set of contiguous segments. The links are *directed* and connect exactly two nodes, one for the origin and one for the destination. This definition can, however, cause some problems when the bi-directional connections must have attributes. The most obvious example is the modeling of a door with an open/close state. To solve this problem, the concept of the *link group* is introduced. It is a set of connections whose geometric representation is the same, and that have shared attributes. Each link can contain one or more constraints that limit their use by the agents. This is particularly important to force them to follow a specific path depending on their type (car, pedestrian...) Constraints may also limit the perceptions of the agents when they are attached with a perception link (see below for details on the different types of links), *e.g.* an agent representing a child may not be able to perceive through a link representing a window. Figure 4 provides the class diagram that defines the concepts related to the links.

The concept of a link is specialized in two main types: navigation link and perception. The first type supports the ability of the agents to navigate from one node to another. The second is used by the agent body's sensors to explore the space, and to collect the entities in the fields of perception of the agents.

**Navigation links** There are three categories of navigation links:

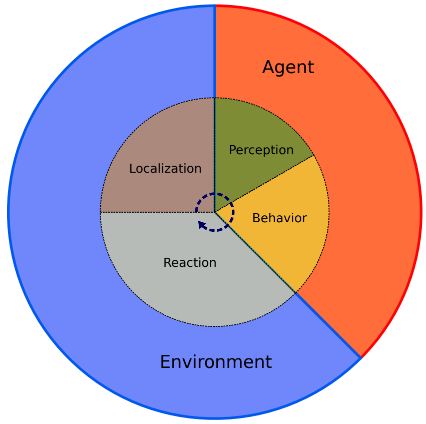


**Fig. 5.** Life-cycle of the simulation, including the environment and the agents

- *Direct navigation link* is the most general and most-used navigation link. It allows the navigation between two nodes without condition/constraint.
- *Lane's change link* is used to connect the nodes representing the adjacent ways of a road. The use of this type of link requires the use of a specific algorithm by the driving agents to ensure that they have the ability to move safely in the traffic.
- *Passing link* is similar to the lane change link, except it allows to connect two ways, which are in opposite directions. This link enables the agents to drive temporarily on the opposite-direction road lane to overtake another vehicle.

**Perception Links** There are two types of perception links:

- *Direct perception link* is the most generic link to connect *two adjacent nodes* together. These links are used by the agents' sensors to explore the space, and to extract the entities in the field of perception of the agents.
- *Conflict link* connects two nodes with *no direct connection*, which are overlapping. These are links that allow sensors of an agent, which is located on the source node, to perceive the entities in the destination node even though navigating from one to the other is impossible. For example, these links are used to connect the pedestrian crossings to the road lanes in order to enable the vehicles as well as the pedestrians to perceive each other. They are also used to link the road lanes that are intersecting.

## 3 Dynamics of the Environment

The dynamics of the environment is divided into four phases, and is directly coupled to the execution of the simulator loop: localization, perception, agent

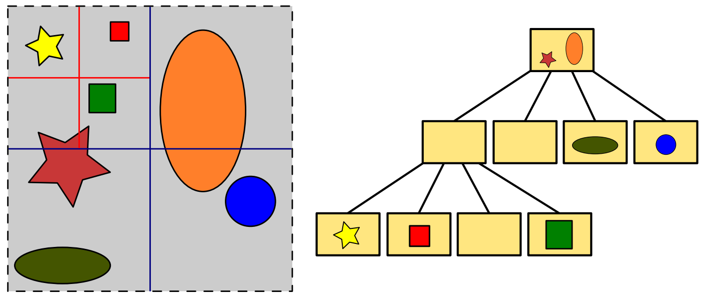

**Fig. 6.** Example of space decomposition with a QuadTree

execution, and reaction. Figure 5 illustrates the life-cycle of the simulation. It is partly based on the influence-reaction model [25].

### 3.1 Localization Phase

The first step is to locate all the entities on the graph so that they can be properly perceived. Static entities or those not having changed of position during a simulation step should be excluded from this process, unless it is the first step of the simulation. The location may be performed sequentially or in parallel using the execution resources provided by the scheduler of the simulator.

The location of the entities on the graph of the environment is divided into two different algorithms. The first is used during the creation of the entity, or, in general, when it has no location information; the other is used when the entity is already recorded on the graph.

In our model, the localization phase is separated from the reaction phase since it is only used to update the navigation/perception graph properly prior to the perception computation. As explained in section 3.3, the reaction can sometimes be handled by a generic physics engine which uses its own specific data structure in place of our graph.

**Initial Localization** When the entity has no location information, *e.g.* when simulation is starting, it makes a request to the environment to get the list of all nodes on which it is located (geometric intersection). The environment must respond very quickly to this request, so as not to slow unduly down the execution of the simulation at each entity creation. For this, several data structures are proposed in the literature: regular grids [26, 27], bounding volume hierarchies [28–30], indexing spatial trees [31], or binary space partition trees [32, 33, 23].

According to the studies mentioned above, the most appropriate structure for our case application is the binary space partition tree, which would be precalculated at the launch time, or during the environment's building phase. The complexity and the execution time associated with this data structure are low for requests, but relatively high for updates. Moreover, it appears that, in practice, the choice of the data structure has only small impacts on the simulator performance, since it is mostly used when creating new entities at launch time. Therefore, we chose to use a regular *QuadTree*[1] which, although theoretically less efficient on average, has the advantage of being much faster to implement and deploy in our simulator. Figure 6 gives an example of a QuadTree. To improve the performances of the QuadTree against the objects that are intersecting the separation lines, *e.g.* the vertical ellipse on Figure 6, 13 child nodes may be added to the tree to support the different cases of intersection of an entity with the separation semi-lines [33, 23].

**Re-Localization** When an entity has location information, they form the basis to search the covered zones in the neighborhood nodes. The difficulty with this algorithm lies in the fact that the navigation graph is cyclic. It is necessary to store the previously visited nodes to avoid unnecessary calculation. The location of an entity in a corridor is based on a binary space partition tree, which is restricted to the space of this corridor. The localization algorithm quickly determines whether the entity intersects the corridor by using the method of the separated axes [34]. The location of an entity in a place is based on the intersection test between the bounding volume of an entity and the geometry of the node. This can be computationally expensive when the geometry of the place is complex, and especially if it has a convex shape. The decomposition of a polygon into zones of influence is used to reduce the number of edges to consider by the algorithm [35].

### 3.2 Perception Phase

The computation of the agent perceptions is not the subject of a specific phase in our model. Indeed, we decided to avoid the overall/global computation of all perceptions, and rather, to calculate the perceptions when an agent is requesting them. The perception becomes an integrated preliminary step for most of the behavior models of the agents. It uses the computation resource, which is assigned to the execution of the agent. This approach enables to restrict the perception computations only when they are needed by the agents. Moreover, paralleling the perception computations is eased with this approach.

### 3.3 Reaction Phase

The reaction phase is related to the influence-reaction model [23, 25]. The agents cannot change directly the state of the environment. In place, they are sending descriptions of changes, named influences, that are gathered by the environment. After all the agents were executed, the environment detects conflicts between

---

[1] The QuadTree is a recursive decomposition of the space into four sub-zones of same size.

influences and computes the resulting reactions. The solving of the conflicts may be a hard task. We propose to use physic-based models for the dynamics of the entities. Consequently, the environment includes a physics engine that can compute all the reactions according to Newton's and mechanic laws. An example of a physic model for a vehicle is given in [36].

## 4 Experiments

Figure 7 illustrates the application of the model for the simulation of pedestrians, bicyclists and drivers in the city of Belfort (France).

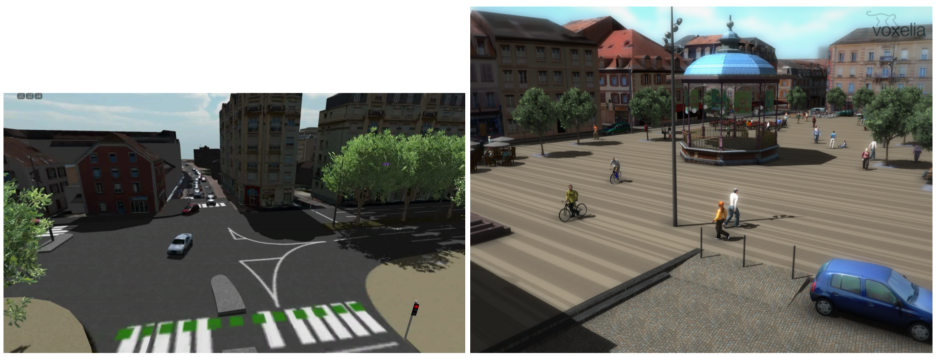


**Fig. 7.** Screenshots of the simulation of the city of Belfort, France

The aim of the project is the enhancement of the public spaces, and to offer to the public a pleasant and comfortable footpath, and an efficient public transport service. This development is part of the urban plan of the city in conjunction with the proposed public transport service OPTYMO[2] [37]. The objectives of the project are to: (i) Give priority to pedestrians with wider sidewalks and crossing off for vehicles; (ii) Limit, by development, the invasion of the car (narrow roads, controlled parking); (iii) Avoid transit traffic, but ensure residential traffic; (iv) Strengthen the tourism and commercial attractiveness of the city.

Our model is implemented on the SIMULATE tool of the Voxelia S.A.S company. It is based on the Unity3D[3] game engine. Each pedestrian or bicyclist uses the force-based model of [24] to determine the best motion direction. Each driver uses an Intelligent Driver Model to determine the acceleration to apply to the car. This acceleration is transformed into a motion direction according to physical model of the car [36]. All the forces given by the agents are treated by the physic engine included in the Unity3D platform, to avoid the collisions. The

[2] `http://www.optymo.fr`
[3] Unity3D Official website: `http://www.unity3d.com/`

results of the simulation are a (confidential) technical report that describes the changes to apply to the city infrastructure, and a collection of videos[4] that are used to explain the changes to the population.

## 5 Conclusion

This paper presents an environment model, which is dedicated to the simulation in 2D or 3D virtual environments. It is primarily designed for the simulation of crowds and traffic. The model combines the agent-oriented approach with object-oriented structures, and the algorithms conventionally used in the fields of virtual reality simulation and video games. Our model is designed to provide the mechanisms of realistic perceptions and actions to agents. It is a software framework that provides a collection of tools to ease the implementation. And it makes faster the simulation of heterogeneous entities in a virtual world (pedestrians, cyclists, drivers. . . ) This simulator has been successfully used in real applications: crowds and traffic in a city, crowds in airport halls [38]. . .

The first major perspective is the scaling of the environment model. We plan to simulate larger populations. Consequently, the impact on the environment model should be studied, and the model adapted, such as applying multilevel simulation [38]. The behaviors of the agents may also be improved, in particular, when the driver wants to change of road lane. Moreover, we assume that the agent architecture is based on three layers: strategy, tactic, operation. The environment model in this paper focuses on the operation level. It should be extended to help the algorithms at the strategy and tactic levels, by introducing ontologies, for example [39].

## Acknowledgments

The model and the experiments are conducted within the commercial tool Simulate of the Voxelia SAS[5] 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. Thalmann, D., Musse, S.R.: Crowd simulation. Springer (2007)
2. Okazaki, S., Matshushita, S.: A study of simulation model for pedestrian movement with evcuation and queuing. In: International Conference on Engineering for Crowd Safety. (1993)
3. Farenc, N., Boulic, R., Thalmann, D.: An informed environment dedicated to the simulation of virtual humans in urban context. In: Proceedings of EUROGRAPHICS 99. (1999) 309–318

[4] Videos are available on: `http://www.multi-agent.fr/Publication:E4MAS14`.

[5] `http://www.voxelia.com`

4. Thomas, G., Donikian, S.: Virtual humans animation in informed urban environments. In: CA '00: Proceedings of the Computer Animation, Washington, DC, USA, IEEE Computer Society (2000) 112
5. Braun, A., Bodmann, B.E.J., Musse, S.R.: Simulating virtual crowds in emergency situations. In: Proceedings of the ACM symposium on Virtual reality software and technology. VRST '05, New York, NY, USA, ACM (2005) 244–252
6. Thompson, P., E., M.: A computer-model for the evacuation of large building population. Fire Safety Journal **24**(2) (1995) 131–148
7. Hareesh, P.: Evacuation simulation: Visualisation using virtual humans in a distributed multi-user immersive vr system. In: VSMM'00. (2000)
8. Klüpfel, H., Meyer-König, M., Wahle, J., Schreckenberg, M.: Microscopic simulation of evacuation processes on passenger ships. Theoretical and Practical Issues on Cellular Automata (2000) 63–71
9. Owen, M., Galea, E., P., L., Filippidis, L.: The numerical simulation of aircraft evacuation and its application to aircraft design and certification. Aeronautical Journal **102** (1998) 301–312
10. Still, G.: Crowd Dynamics. PhD thesis, Warwick University (2000)
11. Szymanezyk, O., Dickinson, P., Duckett, T.: Towards agent-based crowd simulation in airports using games technology. In O'Shea, J., Nguyen, N.T., Crockett, K.A., Howlett, R.J., Jain, L.C., eds.: KES-AMSTA. Volume 6682 of Lecture Notes in Computer Science., Springer (2011) 524–533
12. Pouyanne, G.: Forme Urbaine et Mobilité Quotidienne. PhD thesis, Université Montesquieu - Bordeaux IV (December 2004)
13. Lamarche, F., Donikian, S.: Crowds of Virtual Humans : a New Approach for Real Time Navigation in Complex and Structured Environments. Computer Graphics Forum (Proc. of Eurographics 2004) **23**(3) (2004)
14. Jian, L., Lizhong, Y., Daoliang, Z.: Simulation of bi-direction pedestrian movement in corridor. Physica A: Statistical Mechanics and its Applications **354**(0) (2005) 619 – 628
15. Thalman, D., Farenc, N., Boulic, R.: Virtual human life simulation and database: Why and how. In: DANTE '99: Proceedings of the 1999 International Symposium on Database Applications in Non-Traditional Environments, Washington, DC, USA, IEEE Computer Society (1999) 471
16. Bretagnolle, A., Daudé, E., Pumain, D.: From theory to modelling : urban systems as complex systems. Cybergeo, 13th European Colloquium on Quantitative and Theoretical Geography **335** (September 2003)
17. Odell, J., Parunak, H., Fleisher, M., Brueckner, S.: Modeling Agents and their Environment. In Giunchiglia, F., Odell, J., Weiss, G., eds.: Agent-Oriented Software Engineering III. Volume 2585 of Lecture Notes In Computer Science., N.Y. (USA), Springer-Verlag (2002)
18. Weyns, D., Schumacher, M., Ricci, A., Viroli, M., Holvoet, T.: Environments in multiagent systems. The Knowledge Engineering Review **20**(2) (2005) 127–141
19. Ferber, J., Müller, J.: Influences and reactions : a model of situated multiagent systems. In: Second Internationnal Conference on Multi-Agent Systems (ICMAS). (1996) 72–79
20. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. 2 edn. Pearson Education (2003)
21. Zeigler, B.P., Kim, T.G., Praehofer, H.: Theory of Modeling and Simulation. 2nd edition edn. Academic Press (2000)
22. Barceló, J., ed.: Fundamentals of Traffic Simulation. Volume 145 of International Series in Operations Research & Management Science. Springer (2010)

23. Galland, S., Gaud, N., Demange, J., Koukam, A.: Environment model for multiagent-based simulation of 3D urban systems. In: the 7th European Conference on Multiagent Systems (EUMAS09), Ayia Napa, Cyprus (December 2009) Paper 36.
24. 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 Yasar, A.u.h., Knapen, L., eds.: 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.
25. 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)
26. Joselli, M., Passos, E., Zamith, M., Clua, E., Montenegro, A., Feijo and, B.: A neighborhood grid data structure for massive 3d crowd simulation on gpu. In: Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on. (2009) 121 –131
27. Passos, E.B., Joselli, M., Zamith, M., Clua, E.W.G., Montenegro, A., Conci, A., Feijo, B.: A bidimensional data structure and spatial optimization for supermassive crowd simulation on gpu. Comput. Entertain. **7** (January 2010) 60:1–60:15
28. Yoon, S.E., Manocha, D.: Cache-efficient layouts of bounding volume hierarchies. Computer Graphics Forum **25**(3) (2006) 507–516
29. Gunther, J., Popov, S., Seidel, H.P., Slusallek, P.: Realtime ray tracing on gpu with bvh-based packet traversal. In: Interactive Ray Tracing, 2007. RT '07. IEEE Symposium on. (2007) 113 –118
30. Lauterbach, C., Mo, Q., Manocha, D.: gproximity: hierarchical gpu-based operations for collision and distance queries. Computer Graphics Forum **29**(2) (2010) 419–428
31. Guttman, A.: R-trees: a dynamic index structure for spatial searching. SIGMOD Rec. **14** (June 1984) 47–57
32. Paterson, M., Yao, F.: Efficient binary space partitions for hidden-surface removal and solid modeling. Discrete & Computational Geometry **5** (1990) 485–503 10.1007/BF02187806.
33. Shagam, J., Pfeiffer, J., Jr.: Dynamic irregular octrees (2003)
34. Eberly, D.: Intersection of convex objects: the method of separating axes (2008)
35. Jiménez, J.J., Segura, R.J., Feito, F.R.: Efficient collision detection between 2d polygons. (2007)
36. Gechter, F., Contet, J.m., Lamotte, O., Galland, S., Koukam, A.: Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation. Simulation Modelling Practice and Theory (SIMPAT) **24** (May 2012) 103–114
37. Galland, S., Lamotte, O., Gaud, N.: MetroB: Evaluation and simulation of public transport system. In: the Insitution of Engineering and Technology Conference on Smart and Sustainable City 2011 (ICSSC11), Shanghaï, China, Institute Engineering and Technology and Shanghai University, Shanghaï University Press (July 2011)
38. Demange, J.: A model of environment for the multilevel simulation: application to the simulation of crowds. PhD thesis, Université de Technologie de Belfort-Montbéliard, Belfort, France (December 2012) In French.
39. 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** (January 2014) 64–85