Holonic Multiagent Simulation of Complex Adaptive Systems

Rafik Hadfi(✉) and Takayuki Ito

Department of Computer Science and Engineering,
Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya 466-8555, Japan
{rafik.hadfi,ito.takayuki}@nitech.ac.jp

Abstract. We propose a holonic multiagent simulator that can simulate any complex urban environment. We focus on traffic simulation within any geographic area on earth, subject to any weather conditions. We adopt an agent-based approach for the different beahviors of the vehicles, drivers, and pedestrians. The proposed driving behavioral models can realistically emulate driving behaviors of humans. The resulting simulator can handle all the complexities of such environments in accordance with the laws of physics.

Keywords: Multiagent simulation · Holonic system · Traffic simulation · Geographic information system · Mobility generation · Weather simulation

1 Introduction

Multiagent systems have inspired an increasing number of researchers from different domains. The need for adequate tools for the simulation of complex systems has motivated much of the agent-related research. The main goal of a multiagent system is to model the real world in terms of autonomous agents that can purposely interact with their external environment [18]. An agent can basically gather information from the environment using sensors, while attempting to execute its objectives using effectors [16].

In the context of complex adaptive systems simulation, it is possible to use a multiagent system to reduce the complexity of the simulation by breaking it into several subtasks. For instance, simulating a complex system could be divided into parallel simulations that can handle different subdomains.

Traffic in an open environment is an example of complex adaptive system. It is therefore practical to adopt a multiagent approach for such simulation. Multiagent traffic simulation has been extensively studied since traffic and transportation management require autonomic, collaborative, and reactive agents [1]. It is therefore possible to implement automated traffic control management systems thanks to the fact that agents can operate without centralized control. Furthermore, multiagent systems can connect to distributed subsystems, and can be extended to large-scale multiagent simulation.

J. Bajo et al. (Eds.): PAAMS 2016 Workshops, CCIS 616, pp. 137–147, 2016.
DOI: 10.1007/978-3-319-39387-2_12

In this paper, we propose a multiagent simulator that is suitable for complex adaptive systems. In particular, we focus on simulating traffic as well as the weather conditions that affect its environment. An inherent feature of complex systems is their hierarchical structure and the nested levels of detail that compose them. Therefore, adopting a multiagent agent approach for complex systems modeling will have to acknowledge this organization. Holonic multiagent systems [6] are a practical way for a recursive modeling of autonomous agents, by allowing a dynamic reorganisation of the whole system. Our main motivation is to build a scalable holonic simulator that can emulate traffic as well as any environmental factor that effects traffic flow, routing, and even CO2 emission. Being able to simulate weather is a novel way to approach traffic simulation since it allows the reproduction of real-world scenarios like traffic in natural disaster situations. Such simulator can become a testbed for general-purpose computational intelligence and can be used for the benchmarking of routing algorithms. Additionally, our multiagent approach to behavioral modeling ensures the reproduction of realistic driving behaviors. To this end, we will model our simulator in terms of multiple independent layers. Each layer processes a particular aspect of the simulation through the interactions of its internal elements, or simply, agents. A layer will therefore be represented by a complex network of interacting agents that can communicate within that layer and possibly with other layers. The idea behind our simulation architecture is to adopt a holonic multiagent architecture coupled to a behavioral-based agent simulation. This is in fact a way to refine both the microscopic and the macroscopic aspects found in many traffic simulators. In fact, the availability of data collected via sensors and mobile devices allows us to better model human behaviors. For instance, this allows for the analysis of driver behaviors and the underlying decision-making mechanisms [3,13–15]. As a result, we can build large-scale simulations by embedding the social interactions and by elaborating fine-grained human behaviors [10]. The resulting multiagent social simulation [2] is well suited to any social context due to its ability to simulate pro-active behaviors, parallel computations [12], and dynamic micro scenarios [7,9]. It is within this perspective that we propose our holonic multiagent simulator.

The paper is structured as following. In Sect. 2, we provide the architecture of the simulator. In Sect. 3, we cover the behavioral models used by the different agents. In Sect. 4, we provide the results. In Sect. 5, we conclude and highlight the future work.

2 Simulator Architecture

2.1 Description

Our idea is to represent the whole complex system as a superposition of layers, as shown in Fig. 1. Each layer operates autonomously with all its internal threads, agents, graphical objects, etc. The holonic aspect resides in the fact that the content of each layer is a hierarchy in itself and that all the components of the layers can interact according to the dynamics of the simulation. For instance, the

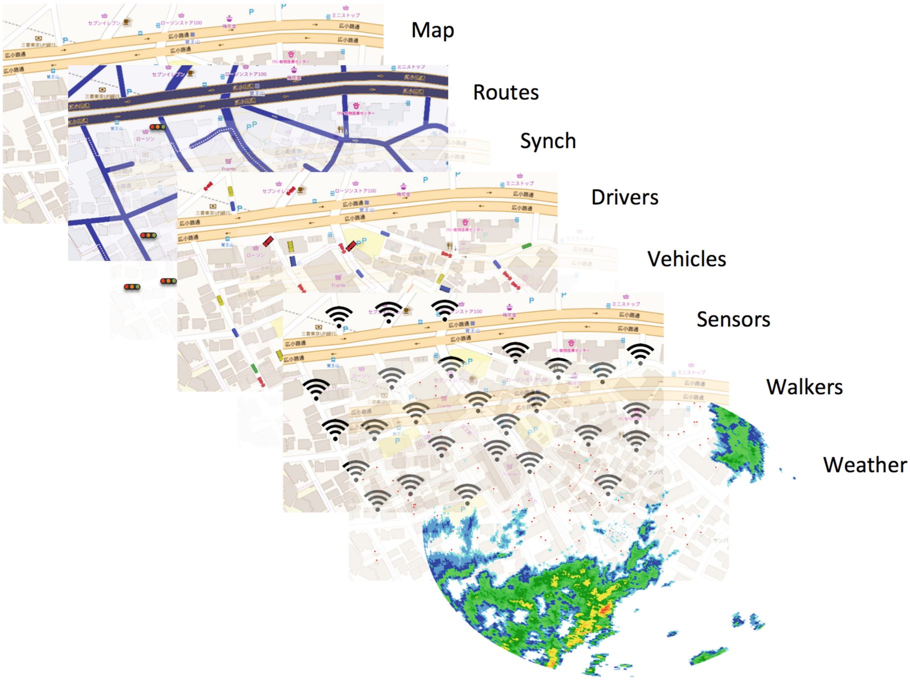


Fig. 1. Multilayered representation of a complex system

vehicles (Vehicles layer) are driven by drivers (Drivers layer) according to traffic lights (Synch layer), while interacting with routes, map, sensors and possibly pedestrians (Walkers). The whole is affected by the weather layer that embodies the physical conditions (precipitations) that affect the whole system.

We note that the layers should be as independent as possible so that the complexity of the simulation does not affect the communication between the different layers. Additionally, there should be a separation between the vehicles and the map in the sense that the way the vehicles navigate their space?should not be?specific to one particular Geographic Information System (GIS). In the following we adopt OpenStreetMap [8] as referential.

2.2 Architecture

The architecture of the simulator is illustrated in Fig. 2. As mentioned in the previous section, we should lower the coupling between the layers of the simulator. Specifically, there should be independence between the actual simulation (**Behavioral Models** and **Physical Engine**) and the corresponding OpenStreetMap rendering. This is important when the renderer is complex, that is, when we are not only rendering vehicles, but rendering pedestrians, weather data, snow, etc. This separation obeys the Dependency Inversion Principle in the sense that physical simulation acts as a high-level, abstract, objective, mathematical

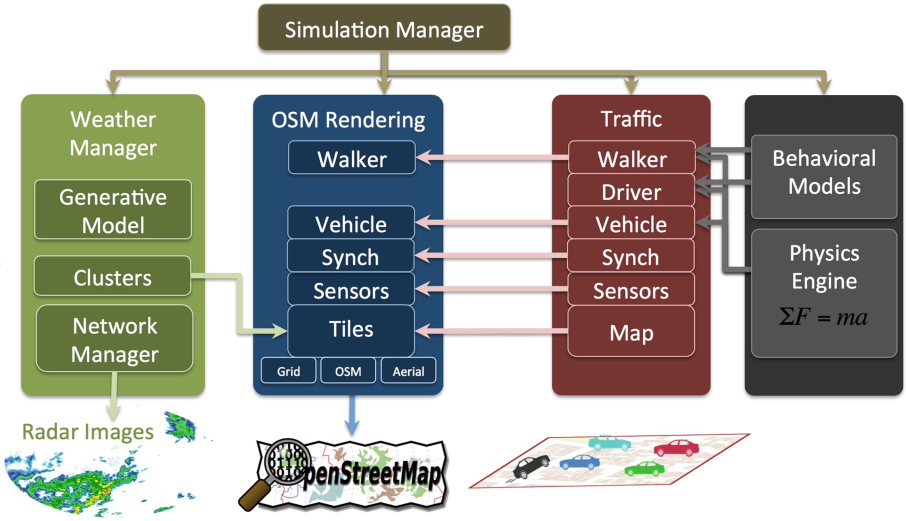


Fig. 2. Architecture

representation of the phenomena we want to simulate, while the OpenStreetMap rendering is one possible way of rendering the simulation. The renderer could easily be exchanged by another one. In the following, we provide a brief description of the components of the architecture.

Physics Engine and Behavioral Models. The Physics Engine component provides an approximate simulation of the physics that underly traffic, motion and the interactions between the agents. This component performs the simulation in real-time before rendering the result. The Behavioral Models describe the scenarios that govern the motion of the vehicles and the pedestrians.

Traffic Tier is composed of the following agent components:

1. **Walker**. Pedestrians are implemented as artificial agents that perform random walks. In particular, an agent performs mobility generation within a closed polygon, or as a random itinerary on the map. Additionally, such agents can predict collisions with other agents.
2. **Driver**. A vehicle is driven by an agent that can also perform collision detection, traffic lights and lanes assessment.
3. **Vehicle**. A vehicle is in fact an agent that modulates the forces acting on a graphical vehicle object. Such forces include the acceleration, direction, location, friction, velocity, and breaking.
4. **Synchronization**. We distinguish a set of agents that update the states of the traffic lights according to the intervals Δ_r, Δ_o, and Δ_g.
5. **Sensors**. A sensor network is built on top of the map and allows the detections of the vehicles motion.

Weather Manager. The weather simulation is in fact the generation of precipitation as if it is detected by a weather surveillance radar (WSR). Additionally, it is possible to download this data in real-time and render it directly onto the map. The weather radars are capable of detecting the motion of rain droplets in addition to the intensity of the precipitation. Both types of data can be analyzed to determine the structure of storms and their potential to cause severe weather.

The simulation of the weather is a mapping between a precipitation vector field onto the geographic tiles, represented as weather clusters. This mapping is computed dynamically since the tiles are loaded as a function of the simulation location.

OpenStreetMap Rendering. The rendering of the geographic map relies on downloading and updating a hierarchy of OpenStreetMap tiles. Such tiles correspond to a specific area of the map and are loaded dynamically, depending on where the simulation is being run. Herein, we use two different maps. The first map (core map) is assigned to the physics engine and is used to run the simulation in real-time so that the result is later rendered onto the second map. The second map represents a real-world referential (OpenStreetMap in our case). The mapping from the core map to the OpenStreetMap map converts absolute references into latitude/longitude references. This conversion is required since it is difficult to manipulate latitudes and longitudes within a small area due to floating-point inaccuracies (E.g. manipulating vectors with latitudes in $\{136.892649, 136.909015, 136.909011\}$). To this end, we use the conversion function (1), where x and y are the coordinates in the core map, x' and y' are the latitude and longitude, w and h are the dimensions of the core map, and x_- (resp. y_-) and x_+ (resp. y_+) are the minimal and maximal latitudes (resp. longitude).

$$x' \leftarrow \frac{x - x_-}{x_+ - x_-} \times (h - 1) \tag{1a}$$

$$y' \leftarrow \frac{y - y_-}{y_+ - y_-} \times (w - 1) \tag{1b}$$

3 Behavioral Modeling

The behavioral models govern the agents mobility as well as the range of actions allowed within the simulator. We distinguish three types of behavioral models for drivers, pedestrians, and vehicles.

3.1 Driving Behavioral Model

The main feature that reflects the driving behavior of a human is the velocity. However, we also need to look at how this velocity changes as function of the turns. In fact, turns are important indicators of the driver's mastery and control of the steering wheel (with angle λ) and its physical effect on the vehicle. We can look at the turning angle θ comprised between the velocity and the direction of the current itinerary. For instance, driving in a straight line corresponds to $\theta = 0$ while a right turn corresponds to $\theta = \pi/2$.

Let us assume that the minimal and maximal velocities of the vehicle are respectively $\boldsymbol{v}^-$ and $\boldsymbol{v}^+$, and that $\boldsymbol{v}$ fluctuates in $[\boldsymbol{v}^-, \boldsymbol{v}^+]$. This interval refers to the driver's spectrum of physically allowed velocities. A behavioral model is therefore a specific way of mapping the turning angle θ to a specific velocity $\boldsymbol{v}$, illustrated in function (2).

$$\boldsymbol{v}_\theta : [0, \pi] \longrightarrow [\boldsymbol{v}^-, \boldsymbol{v}^+] \tag{2}$$

Figure 3 shows three behavioral models. Model v_1 corresponds to what is perceived as a reckless driver since he barely decelerates when performing right turns. Model v_2 shows a conservative driver since he decelerates drastically when reaching right turns. Finally, v_3 shows a standard driver. We assume that a behavior is invariant beyond π (U-turn), and thus, choose to limit the angular interval to $[0, \pi]$.

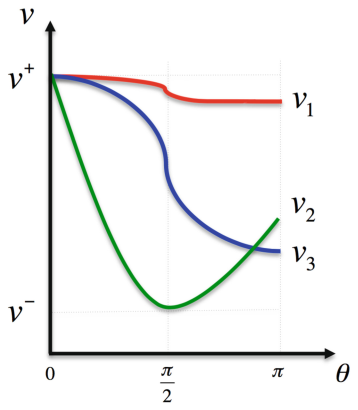


Fig. 3. Behavioral models $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$

3.2 Pedestrian Behavioral Model

A pedestrian (walker) is modeled as an agent that moves within a closed polygonal space. The polygonal space is in fact a block on the map.

3.3 Vehicle Behavioral Model

For a vehicle, the act of moving from point A to point B is reduced to the execution of set of operations that could be summarized as following.

1. Updating the forces that act upon the vehicle.
2. Updating the acceleration $\boldsymbol{a}$.
3. Updating the velocity $\boldsymbol{v}$.
4. Updating the direction $\boldsymbol{d}$.
5. Updating the location $\boldsymbol{x}$ of the vehicle.

These actions are described as following:

Updating direction $\boldsymbol{d}$ according to (3). Here, λ is the steering wheel angle, L is the vehicle wheelbase, t_r is the turning radius, u_m is the number of units per meter, p is the simulation step, $\alpha_{\boldsymbol{d}}$ is the angle of direction $\boldsymbol{d}$, and α_s is the angular speed.

$$t_r \leftarrow L/sin(\lambda \times \frac{\pi}{3}) \tag{3a}$$

$$\alpha_s \leftarrow (\|\boldsymbol{v}\| \times u_m)/t_r \tag{3b}$$

$$\boldsymbol{d} \leftarrow \alpha_{\boldsymbol{d}} + \alpha_s \times p \times \frac{180}{\pi} \tag{3c}$$

Updating velocity $\boldsymbol{v}$ according to (4), with $\boldsymbol{a}$ being the acceleration.

$$\boldsymbol{v} \leftarrow \boldsymbol{d} \times \|\boldsymbol{v}\| + \boldsymbol{a} \times p \tag{4}$$

In case $|\alpha_{\boldsymbol{v}} - \alpha_{\boldsymbol{d}}| > \frac{\pi}{2}$, $\boldsymbol{v}$ is nullified. $\alpha_{\boldsymbol{v}}$ being the angular velocity.

Setting direction as function of θ as in (5).

$$\theta' \leftarrow \theta \times \frac{\pi}{180} \tag{5a}$$

$$\boldsymbol{d} \leftarrow \big(\cos(\theta'),\ \sin(\theta')\big) \tag{5b}$$

Updating acceleration based on Newton's second law ($\sum \boldsymbol{f} = m\boldsymbol{a}$) as in (6). The motive force of the vehicle is $\boldsymbol{F}_m$, the friction force is $\boldsymbol{F}_f$, and m is the vehicle's mass.

$$\boldsymbol{a} \leftarrow \frac{\boldsymbol{F}_m + \boldsymbol{F}_f}{m} \tag{6}$$

Updating all the forces, which requires updating the friction force whenever the driver breaks. This is done according to (7). $\boldsymbol{a}_{max}$ is the maximal acceleration.

$$\boldsymbol{F}_f \leftarrow \boldsymbol{d} \times -\boldsymbol{a}_{max} \tag{7}$$

Updating the location x according to (8).

$$x \leftarrow x + \boldsymbol{v} \times u_m \times p \tag{8}$$

4 Results

The interface of the simulator is shown in Fig. 4. The default view is the OpenStreetMap view. However, it is possible to show different perspectives that illustrate the superposed weather and traffic simulations as shown in Fig. 5. The user can macroscopically switch between the views (Aerial view, OpenStreetMap view, Grid view) of the simulation by altering the opacity of the layer.

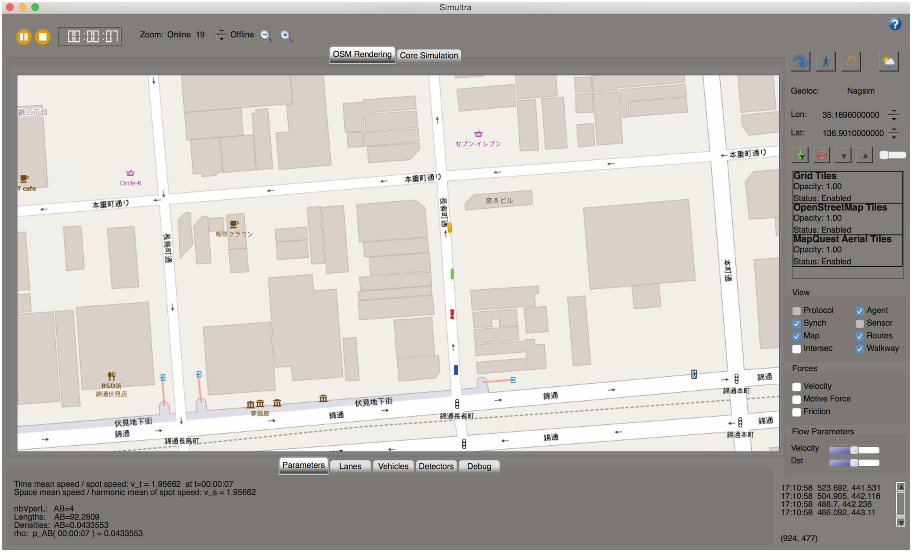
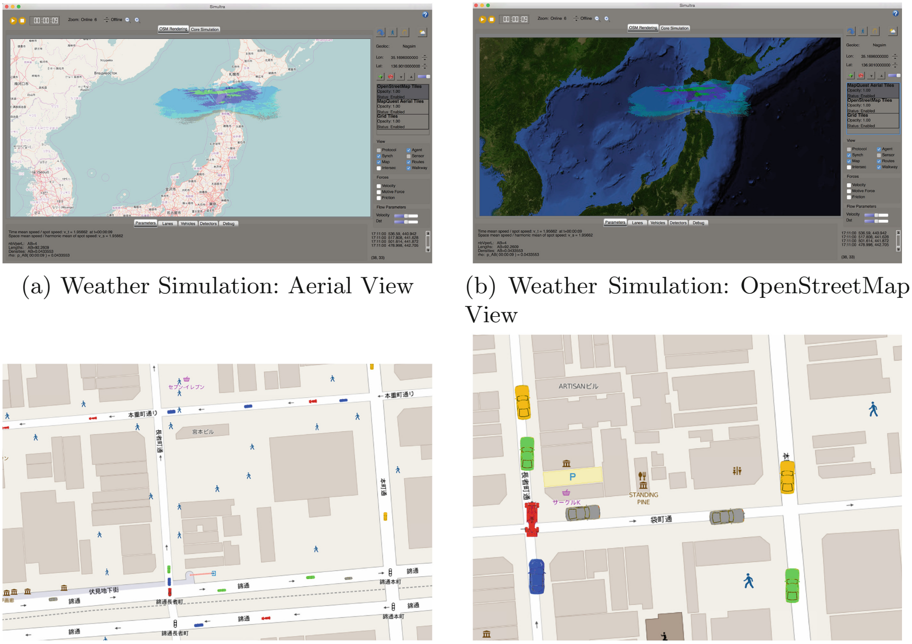


Fig. 4. Simulator main interface

The separation between the layers allows the simulations to be scalable in the number of vehicles and pedestrians, despite the complexity of their behaviors. Furthermore, running the physical simulations within one monolithic components allows us to render all the results faster than if each layer had to perform complex physical computations separately. The simulator can additionally generate rich datasets relative to the vehicles, the pedestrians, the drivers' actions, and the weather effect of the simulation map.

5 Related Work

In the context of traffic simulation, the multiagent approach is well suited to build traffic simulators and reproduce the dynamic and complex phenomena that are hard to express using purely mathematical models. In the mathematical approach, we tend to reproduce the vehicles streams from car-following dynamics obtained empirically from data collected at different operating road sections [11].



(a) Weather Simulation: Aerial View

(b) Weather Simulation: OpenStreetMap View

(c) Macroscopic Traffic View

(d) Microscopic Traffic View

Fig. 5. Weather and traffic views

However, such mathematical approaches are only suited for simple traffic analysis since they are not realistic and do not offer accurate traffic flow for high density traffic.

As opposed to this mathematical approach, we find the behavioral approach, which is more coherent with the multiagent paradigm. Such approach relies on the interaction between various agents, such as vehicles, drivers, pedestrians, traffic lights, sensors, and so forth. The simulation is therefore an emergent phenomena of all these interactions. An example of such multiagent behavioral simulator is Archsim [5]. Such simulator allows a realistic coupling between the driving simulation and the traffic simulation. Another similar approach attempts to realistically model road junctions [4]. This approach is based on opportunistic individual behaviors that can detect critical circumstances.

Another family of simulators rely on Stochastic Cell Transmission Model (SCTM) [17], and has been proposed as a macro traffic simulation model of high accuracy. SCTM can represent the uncertainty of the traffic states as well as the changing travel demand and supply conditions. So far, SCTM has only been applied to freeways and simple networks that are one-to-one origin-destination networks.

6 Conclusion

We proposed a holonic multiagent simulator that can reproduce traffic as well as the weather conditions of the underlying geographic area. The proposed architecture ensures a low coupling of the different hierarchies of the multiagent system, which allows a realistic reproduction of traffic as well as the behaviors of the drivers and pedestrians.

As future directions, we think of distributing the multiagent system so that the different simulations are assigned to different clusters. This would allow us to add more computational intelligence by improving the behavioral models of the pedestrians and adding argumentation between the vehicles. Another direction is to adopt a 3D renderer instead of the 2-dimensional OpenStreetMap rendering. Most importantly, we think of deploying different routing algorithms on the simulator and evaluating the evolution of traffic congestion.

Acknowledgement. This work has been partially supported by the project “Multi-agent Future Traffic Prediction to Relieve Traffic Congestion” in the issue “Research and Development on Applications of Social Big Data”.

References

1. Chen, B., Cheng, H.: A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transp. Syst. **11**, 485–497 (2010)
2. Davidsson, P.: Multi Agent Based Simulation: Beyond Social Simulation. Springer, Heidelberg (2001)
3. Dia, H.: An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. Transp. Res. Part C: Emerg. Technol. **10**(5–6), 331–349 (2002)
4. Doniec, A., Mandiau, R., Piechowiak, S., Espie, S.: A behavioral multi-agent model for road traffic simulation. Transp. Res. Part C: Emerg. Technol. **21**(8), 1443–1454 (2011)
5. Espie, S.: Archisim, multi-actor parallel architecture for traffic simulation. In: Proceedings of the Second World Congress on Intelligent Transport Systems, vol. 4 (1995)
6. Fischer, K., Schillo, M., Siekmann, J.H.: Holonic multiagent systems: a foundation for the organisation of multiagent systems. In: Mařík, V., McFarlane, D.C., Valckenaers, P. (eds.) HoloMAS 2003. LNCS (LNAI), vol. 2744, pp. 71–80. Springer, Heidelberg (2003)
7. Gilbert, N.: Computer Simulation of Social Processes. Number 6. Social Research Update, Department of Sociology, University of Surrey, UK (1994)
8. Haklay, M.M., Weber, P.: Openstreetmap: user-generated street maps. IEEE Pervasive Comput. **7**(4), 12–18 (2008)
9. Harding, A.: Microsimulation and Public Policy. Elsevier, Amsterdam (1996)
10. Hattori, H., Nakajima, Y., Yamane, S.: Massive multiagent-based urban traffic simulation with fine-grained behavior models (2011)
11. Lieberman, E., Rathi, A.: Traffic flow theory. Trans. Res. Part C: Emerg. Technol. (1997)

12. Misra, J.: Distributed discrete-event simulation. ACM Comput. Surv. (CSUR) **18**, 39–65 (1986)
13. Nagel, K., Marchal, F.: Computational methods for multi-agent simulations of travel behaviour. In: Proceedings of International Association for Travel Behavior Research (IATBR), Lucerne, Switzerland (2003)
14. Rossetti, R.J., Bordini, R.H., Bazzan, A.L., Bampi, S., Liu, R., Vliet, D.V.: Using BDI agents to improve driver modelling in a commuter scenario. Trans. Res. Part C: Emerg. Technol. **10**(5–6), 373–398 (2002)
15. Rossetti, R.J., Liu, R., Cybis, H.B., Bampi, S.: A multi-agent demand model. In: Proceedings of the 13th Mini-Euro Conference and the 9th Meeting of the Euro Working Group Transportation, pp. 193–198, Bari, Italy (2002)
16. Russell, S., Norvig, P.: Artificial intelligence: a modern approach (1995)
17. Sumalee, A., Zhong, R., Pan, T., Szeto, W.: Stochastic cell transmission model (SCTM): a stochastic dynamic traffic model for traffic state surveillance and assignment. Trans. Res. Part B: Methodol. **45**(3), 507–533 (2011)
18. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(02), 115–152 (1995)