Chapter 24
MOSAIIC: City-Level Agent-Based Traffic Simulation Adapted to Emergency Situations

Guillaume Czura, Patrick Taillandier, Pierrick Tranouez, and Éric Daudé

Abstract In this paper, we present MOSAIIC, an agent-based model to simulate the road traffic of a city in the context of a catastrophic event. Whether natural (cyclone, earthquake, flood) or human (industrial accident) in origin, catastrophic situations modify both infrastructures (buildings, road networks) and human behaviors, which can have a huge impact on human safety. Because the heterogeneities of human behaviors, of land-uses and of network topology have a great impact on the traffic flows, the agent-based modeling is particularly adapted to this subject. In this paper, we focus on the new traffic model itself: the way geographical data is used to build a network, the various behaviors of our agents, from the individual to the collective level.

24.1 Introduction

Nowadays, traffic simulations are often used by urban planners to make decisions concerning road infrastructures. Many models have been developed these last years. These models are grouped according to their levels of representation: macroscopic [1], mesoscopic [2], microscopic [3] and nanoscopic [4].

A modeling approach that is particularly well-fitted for micro-simulation is agent-based modeling. It allows to consider the heterogeneity of driver behaviors and to take into account the global impact of local processes.

Such approach is increasingly used as many frameworks allowing urban planners to easily build their own scenarios (MATSIM [5], SUMO [6], AgentPolis [7]) are

G. Czura (✉) • P. Taillandier
UMR CNRS IDEES 6266, University of Rouen, Mont-Saint-Aignan, France
e-mail: guillaume.czura@gmail.com; patrick.taillandier@univ-rouen.fr

P. Tranouez
EA LITIS 4108, University of Rouen, Saint-Étienne-du-Rouvray, France
e-mail: pierrick.tranouez@univ-rouen.fr

É. Daudé (✉)
CNRS, UMIFRE 20 CSH, CNRS-MAE, Delhi, India
e-mail: eric.daude@univ-rouen.fr

H. Takayasu et al. (eds.), *Proceedings of the International Conference on Social Modeling and Simulation, plus Econophysics Colloquium 2014*, Springer Proceedings in Complexity, DOI 10.1007/978-3-319-20591-5_24

developed. However, if these frameworks are often well-adapted to traffic in normal condition, very few tools allow the simulation of uncommon events such as natural or technological hazards. Actually, in this context, being able to simulate the traffic in a realistic way while taking into account the road infrastructure (crossing, traffic signals...), the properties of the cars (length, max speed...) and the personality of the drivers (tendency to respect the norms) is mandatory. Most other frameworks work at a higher level, supposing regularities and statistical behaviors. But in a disaster individual decisions can lead to important collective consequences. Two drivers can leave their car, thus blocking hundred behind them. Drivers may react to things they see, fleeing by taking one-way streets in reverse, creating a jam in the road leading to this street. Individual-based modeling and micro-simulation are the only way to incorporate those possibilities, not just origin destination matrices and shortest path algorithms. For modelers without high level programming skills, adapting these platforms to specific application contexts is out of reach as they require to write code in JAVA or C++. As a result, many simulators are still developed from scratch or with a generic platform (e.g. [8–10]).

In this paper, we propose a new generic model dedicated to traffic simulation based on the work of [10–12] called MOSAIIC. This model, which have been implemented using the GAMA modeling platform [13], is easily tunable through a specific modeling language. Moreover, this model manages road infrastructures and traffic signals, input from real geographical data, as well as a detailed implementation of cars and drivers: choice of destination, acceleration and deceleration according to the surrounding traffic and the regulation, lanes changing, crossroads crossing etc. In addition, it allows to take into account the personality of each driver: respect of norms (traffic light, right of way, speed limits...), the management of tailgating.

The paper is organized as follows: Sect. 24.2 is dedicated to the presentation of the generic MOSAIIC Agent-based model. Section 24.3 concludes and presents some perspectives.

24.2 The MOSAIIC Agent-Based Traffic Model

As stated in the previous section, we chose to implement the model with the GAMA platform. The GAMA platform provides modelers—who quite often are not developers—with tools to develop highly complex models. In particular, it offers a complete modeling language (GAML: GAma Modeling Language) and an integrated development environment that allows modelers to quickly and easily build models. Indeed, the GAML language is as simple to use and to understand as the Netlogo modeling language [14] and does not require high level programming skills. In addition, GAMA provides different features that can be used by modelers to develop traffic models. In particular, GAMA allows to simply load GIS data (shapefiles, OSM data...), to define graphs from polyline geometries, to compute

shortest paths and to move agents on a polyline networks. At last, it integrates an extension dedicated to fine-scale traffic simulation [15].

24.2.1 Structure of the Network

A key issue for our model is to be versatile enough to be usable with most of classic road GIS data, in particular OSM[1] data. We choose then to use a classic format for the roads and nodes (Fig. 24.1). Each road is a polyline composed of road sections (segments). Each road has a target node and a source node. Each node knows all its input and output roads. A road is considered as directed. For bidirectional roads, 2 roads have to be defined corresponding to both directions. Note that for some GIS data, only one road is defined for bidirectional roads, and the nodes are not explicitly defined. In this case, it is very easy, using the GAML language, to create the reverse roads and the corresponding nodes (it only requires few lines of GAML).

A road can be composed of several lanes (Fig. 24.2). The vehicles are able to change at any time its lane and even use a lane of the reverse road. In this case, the vehicle "cross" the road (for example going from Road 2 to Road 1 in the Fig. 24.1). Legal speed is another property of the modeled road. Note that even if the user of the model has no information about these values for some of the roads (the OSM data are often incomplete), it is very easy using the GAML language to fill in the missing values by a default values. It is also possible to change these values dynamically

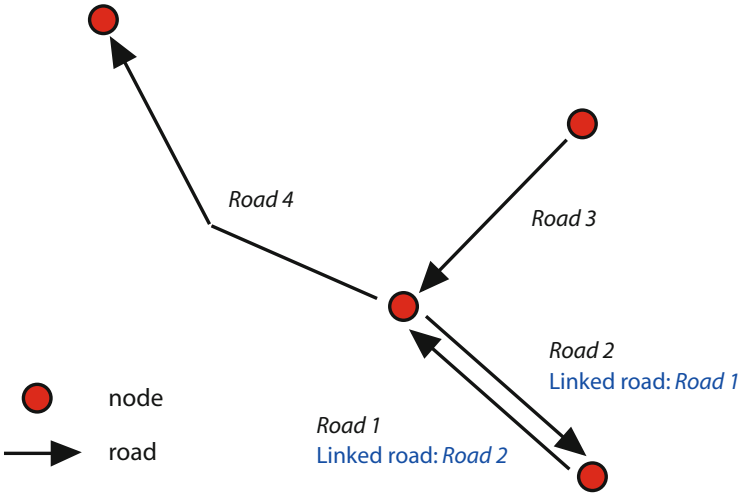

Fig. 24.1 Roads and nodes description in the model

[1] OSM: OpenStreetMap.

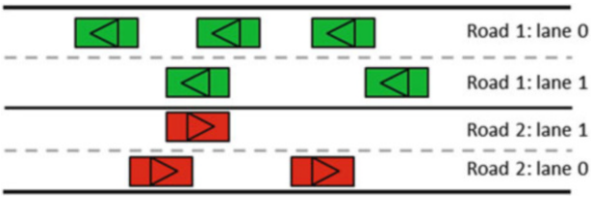

Fig. 24.2 Roads and lanes description in the model

during the simulation (for example, to model that after an accident, a lane of a road is closed or that the speed of a road is decreased by the authorities).

In order to give the modelers the possibility to simply add dynamics to these infrastructures (e.g. to add a deterioration dynamic to roads), we chose to represent all the road infrastructures (road, traffic signals) as agents.

For each roads, a list of predefined variables is defined. Some of them are linked to the road properties:

- *lanes*: number of lanes.
- *maxspeed*: maximum authorized speed on the road.

In the same way, for each nodes, a list of predefined variables is defined. Amongst them, the most important is the list of stop signals, and for each stop, the list of roads concerned by it.

The complete list of variables for roads and nodes can be founded in [15].

24.2.2 Driver Agents

Concerning the driver agents, we propose a driving model based on the one proposed by Tranouez et al. [10]. Each driver agent has a planned trajectory that consists in a succession of edges. When the driver agent enters a new edge, it first chooses its lane according to the traffic density, with a bias for the rightmost lane. The movement on an edge is inspired by the Intelligent Driver Model [16]. A difference with our driving model is that in our model the drivers have the possibility to change their lane at any time (and not only when entering a new edge). In addition, we have defined more variables for the driver agents in order to give more possibilities for the modelers to tune the driver behavior.

The driver agents have several variables that will define the car properties and the personality of the driver, ranging from the length of the vehicle to the probabilities of respecting right of way. The values of these variables can be modified at any time during the simulation. For example, the probability to take a reverse road can be increased if the driver is stuck for several minutes behind a slow vehicle.

24.2.3 Dynamics of the Model

One step of the simulation represents 1 s. The dynamics of the model is based six consecutive steps:

1. Each road agent computes the potential traffic jams
2. Each traffic signal computes its new state.
3. New drivers arrive in the simulation
4. Drivers that do not have a path to reach their destination (or that should recompute them owing to changes in their context) compute it.
5. Drivers drive toward their final target. Note that the driving step is asynchronous. agents move one after the other. The order of activation of the driver Agents depend on their distance to the end of their current road: the drivers closer to the road end are activated first.
6. Drivers that reach their final target are removed from the simulation

24.2.3.1 Traffic Jam Management

Each road has the capability to compute the traffic jams on it. A traffic jam is defined as the presence on the road of at least *number_threshold* drivers of which the speed is inferior to *speed_threshold*. The *number_threshold* variable depends on the capacity of the road (will be lower for a small road than for a long road) and the *speed_threshold* variable on the *max_speed* on the road. A traffic jam becomes real for drivers if it exists for at least *time_threshold*.

24.2.3.2 Traffic Signal Update

Each traffic signal (only traffic light in our model) update its state counter.

24.2.3.3 New Driver Arrival

According to the current time and the data provided, a certain number of drivers are created. We assume that the model user has data (scenario) concerning the number of drivers departing at each period of time (for instance, there are often more drivers departing at 8AM than at 11PM). These drivers are located on one of the nodes according to the scenario data. Indeed, we assume here that the user has also data concerning the Origin and Destination of drivers. According to this data, a probability to use each node as an origin is computed and used to define the initial location of each new driver. In the same way, a probability to use each node as a destination according to a given origin is computed and used to define the final target of each new driver.

24.2.3.4 Computation of the Path

A driver can compute the path between its current location and its final target using a graph structure (each road will be an edge of the graph). In order to do so, the driver will use its own weights concerning for the edge. We defined four profiles of driver (i.e. four types of weight):

- minimize the travel distance
- minimize the travel time
- minimize the travel time and favorise roads with many lanes
- minimize the travel time and avoid traffic lights

In addition, when a driver perceived that its path will cross at least one known uncommon event (traffic jam, blocked roads), it will test a the *proba_avoid_event* probability to define if it will try to avoid it or not. If it tries to avoid it, it has two specific behaviors that will depends on the test of the *proba_know_map* probability. If the driver knows the map, it will compute a new shortest path taking into account all the information it has concerning uncommon events. In the other case, it will just try to choose through heuristics roads without uncommon events that will allow it to move closer to its final target.

24.2.3.5 Driving Step

The driving action of the driver agents work as follow: while the agent has the time to move, it first defines the speed he tries to reach based on different variables. Then, the agent moves toward the current target and computes the remaining time (Fig. 24.3). More specifically, each driver has a remaining time which is initially set to 1 s. Remaining time decreases after it drives, and it can continue to drive until remaining time becomes 0 or it has to stop at the intersection.

During the movement, the agents can change lanes (see below). If the agent reaches its final target, it stops; if it reaches its current target (that is not the final target), it tests if it can cross the intersection to reach the next road of the current path. If it is possible, it defines its new target and continues to move. The function that defines if the agent crosses or not the intersection to continue to move works as follow (Fig. 24.4): first, it updates its known uncommon events by adding all the uncommon events it perceives (the ones at a distance lower than its *perception_distance)*. If its current path crosses an uncommon event, it will apply its path computation action. After that, it tests if the road is blocked by a driver at the intersection (if the road is blocked, the agent does not cross the intersection). Then, if there is at least one stop signal at the intersection (traffic signal, stop sign. . .), for each of these signals, the agent tests its probability to respect or not the signal (note that the agent has a specific probability to respect each type of signals). If there is no stopping signal or if the agent does not respect it, the agent checks if there is at least one vehicle coming from a right (or left if the agent drives on the left side) road at a distance lower than its security distance (i.e. minimal distance to the closest vehicle

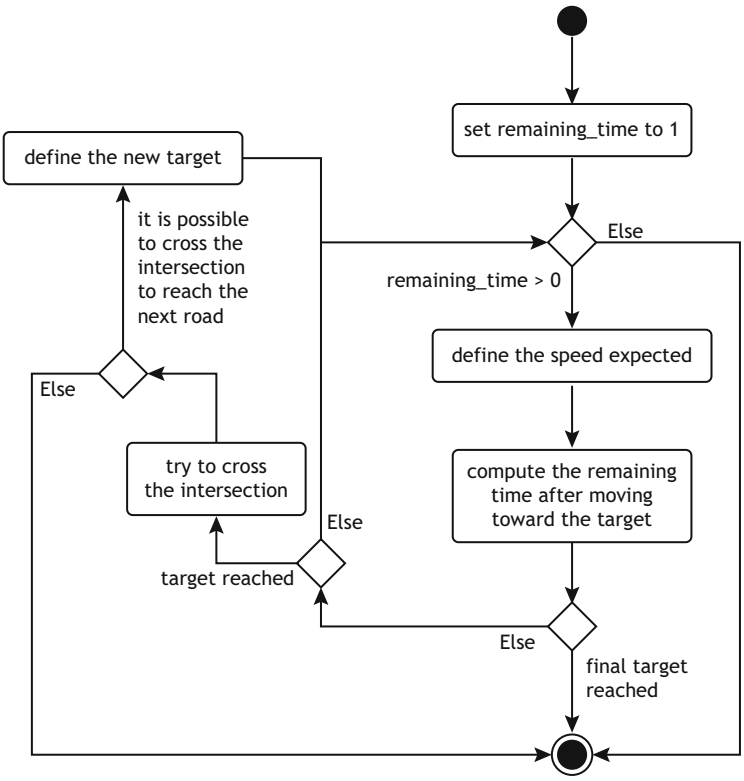

Fig. 24.3 Agent driving action algorithm

from which the agent feels safe). If there is one, it tests its probability to respect this priority. If there is no vehicle from the right roads or if it chooses to do not respect the right priority, it tests if it is possible to cross the intersection to its target road without blocking the intersection (i.e. if there is enough space in the target road). If it can cross the intersection, it crosses it; otherwise, it tests its probability to block the node: if the agent decides nevertheless to cross the intersection, then the perpendicular roads will be blocked at the intersection level (these roads will be unblocked when the agent is going to move).

Concerning the movement of the driver agents on the current road, the agent moves from a section of the road (i.e. segment composing the polyline) to another section according to the remaining time and to the maximal distance that the agent can moves (Fig. 24.4). For each road section, the agent first computes the initial remaining distance it can travel according the remaining time and its speed (i.e. max distance it can travel if there is no other vehicle). Then, the agent computes its security distance (i.e. minimal distance to the next vehicle from which the agent feels safe) according to its speed and its *security_distance_coeff*. While its remaining distance is not null, the agent computes the maximal distance it can travel

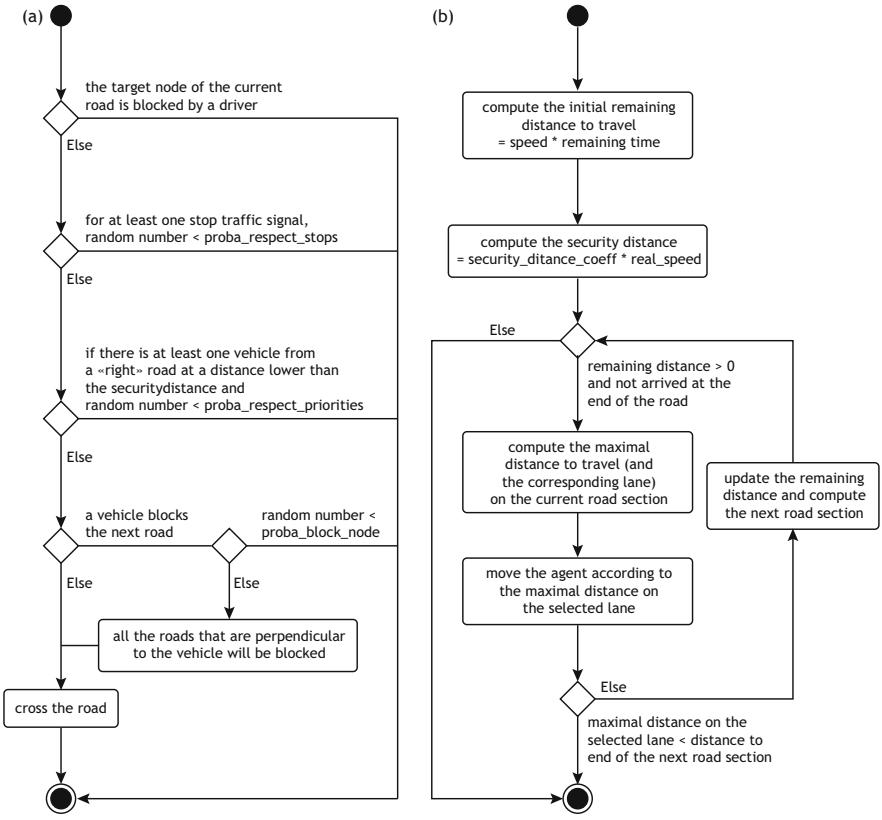

Fig. 24.4 (**a**) Crossing of an intersection (case where the driver drives on the right side of the road) and (**b**) Move on the current road algorithms

(and the corresponding lane), then it moves according to this distance (and updates its current lane if necessary). If the agent is not blocked by another vehicle and can reach the end of the road section, it updates its current road section and continues to move. The agent changes lanes if it computes it could go further in its time slot on another lane.

Figure 24.5 shows a snapshot of a simulation carried out for the agglomeration of Rouen (France).

24.3 Conclusion

In this paper, we presented a new generic traffic model. From this model, which was implemented with the GAMA platform, traffic simulations with a detailed representation of the driver operational behaviors can be built. In particular, it

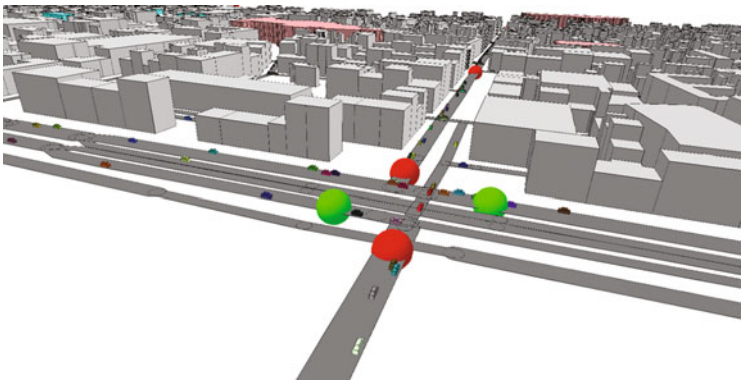

Fig. 24.5 MOSAIIC traffic model applied to Rouen agglomeration, France

models the road infrastructures and traffic signals, the lane changes of the drivers and their respect of norms. In comparison to the use of existing traffic simulation frameworks, the advantage of our model is to enable modelers to easily define models adapted to their application context.

We plan to enrich the model in order to make the driver agents more cognitive, in particular concerning their choice of path and their adaptation to the their current context in emergency situation. For this, we plan to give the driver agents a BDI architecture that can be based on [17, 18].

If the model is already capable to simulate tens of thousands of driver agents, we plan to improve its efficiency by using High Performance Computing and in particular distribution on GPU to enable large scale simulations with millions of driver agents.

At last, we plan as well to develop new tools to help people to prepare their data. The goal will be to offer the possibility from incomplete OSM data to automatically fill the missing attributes, and to create a consistent network (with its infrastructure and traffic signals). A particular attention will be brought on traffic signals and traffic lights.

Acknowledgements This research was founded by the project MOSAIIC, funded by the Region Haute-Normandie, the Large-Scale Research Network TERA and the Technological Risk Control Network (TRCN).

References

1. Kotsialos A, Papageorgioy M, Diakaki C, Pavlis Y, Middleham F (2002) IEEE Trans Intell Transp Syst 3(4):282
2. Waldeer K (2004) Transp Theory Stat Phys 33(1):31
3. Miller JE, Hunt DJ, Abraham JE, Salvini PA (2004) Comput Environ Urban Syst 28:9
4. Daiheng N (2003) Intelligent vehicles symposium, pp 47–52
5. Balmer M, Rieser M, Meister K, Charypar D, Lefebvre N, Nagel K, Axhausen K (2009) Multi-agent systems for traffic and transportation engineering. Information Science Reference, Hershey, pp 57–78
6. Krajzewicz D, Erdmann J, Behrisch M, Bieker L (2012) Int J Adv Syst Meas 5(3&4):128
7. Jakob M, Moler Z (2013) Proceedings of the 16th IEEE intelligent transportation systems conference (ITSC 2013)
8. Horn M (2002) Transp Res A Policy Pract 36(2):167
9. Banos A, Marilleau N, Team M (2012) ECSS
10. Tranouez P, Daudé E, Langlois P (2012) J Nonlinear Syst Appl 3(2):98
11. Daudé É, Provitolo D, Dubos-Paillard E, Gaillard D, Eliot E, Langlois P, Zimmermann E, Saint-Gérand T 2009) From system complexity to emergent properties. Springer, Berlin/Heidelberg, pp 165–178
12. Daudé É, Tranouez P, Langlois P (2009) The 3rd international conference on complex systems and applications, pp 116–121
13. Grignard A, Taillandier P, Gaudou B, Vo D, Huynh N, Drogoul A (2013) PRIMA 2013: principles and practice of multi-agent systems. Lecture Notes in computer science, vol 8291. Springer, Berlin, pp 117–131
14. Wilensky U (1999) Netlogo. Technical report. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://www.ccl.northwestern.edu/netlogo/
15. Taillandier P (2014) Eighth international workshop on agents in traffic and transportation (ATT)
16. Kesting A, Treiber M, Helbing D (2007) J Transp Res Board 1999:86
17. Taillandier P, Therond O, Gaudou B (2012) International environmental modelling and software society (iEMSs)
18. Le VM, Gaudou B, Taillandier P, Vo DA (2013) Advanced methods and technologies for agent and multi-agent systems (KES-AMSTA). Frontiers in artificial intelligence and applications, vol 252. IOS Press, Amsterdam, pp 395–403