A versatile description framework for modeling behaviors in traffic simulations

Alexandre Bonhomme, Philippe Mathieu, Sébastien Picault

A versatile description framework for modeling behaviors in traffic simulations

Alexandre Bonhomme, Philippe Mathieu, Sébastien Picault
LIFL UMR CNRS 8022
Lille 1 University
Villeneuve d'Ascq, France
philippe.mathieu@univ-lille1.fr

***Abstract*—Microscopic simulations of road traffic are a typical application domain for Multi-Agent Systems. Indeed, the individual-based approach allows to take into account the diversity of behaviors so as to consider real situations. More recently, geographical databases provide environmental information under open formats, which offers the opportunity to design agent-based traffic simulators which can be continuously informed of changes in traffic conditions. The use of such data, together with the adaptability of MAS, allows the realization of decision support systems that are able to integrate environmental and behavioral modifications in a direct way, and compare various scenarios built from different hypotheses in terms of actors, behaviors, environment and flows. We describe here a modeling approach and a comprehensive process which lead to the development of such a tool.**

***Keywords*-Multi-Agent Simulation; GIS; Road Traffic; Traffic Generator; Decision Support System**

I. Introduction

Traffic simulation is a domain where the individual-based approach proved very early its benefits [1], [2], especially to evaluate the impact of individual decisions on the macroscopic state of road traffic or, more recently, on the capacity to integrate a statistical variability in norm violation by drivers [3].

One major difficulty in designing a traffic simulator is the diversity of goals: flux study, binnacle ergonomics, local effect of roads planning modifications on drivers behavior... Concerned space and time scales varies considerably and leads in general to *ad hoc* models (i.e. designed for a particular usage) which are hard to revise. In fact, in a domain like transportation, this multiplicity of goals and scales might rather be taken into account as a general context for the design of the simulation tool, in order to answer really different questions with a homogeneous platform composed of easily modifiable models from explicit knowledge. Thus, it is possible to build a library of behavioral models that could be reused in different contexts and for different purposes, selected according to simulation hypotheses and usage scenarios.

Furthermore, in order to ensure maximum reliability in simulation results, it is necessary to configure the environment and agent behaviors on the basis of real data. Regarding agent behavior, we have already proposed methods in this direction for transports [3] and others domains [4]. We are going to show here that behaviors can be coupled with an environment built from cartographic data, and parameterized by (or compared with) road flux information.

In the following, we present in particular *TrafficGen*, an experimental traffic simulator that we have developed as proof of concept for assessing our approach. It does not aim currently at competing in performance or in quality with professional, specialized tools like ArchiSim, SCANeR II or Synchro Studio, but rather at determining how to build a decision support tool which allows a simple continuous update of data, as well as an easy revision of models and comparison of scenarios.

This article is organized as follows. First, we give an overview of available open data, their sources and structures. Then we consider the state of the art in traffic simulation, and discuss the workflow we define for acquiring, filtering, structuring data, integrating them in *TrafficGen* and finally analyzing the simulation outputs. We also explain how to build a multi-agent model which is modular, easily extensible and reviewable. We illustrate our approach on a simple experiment, before concluding about the perspectives of this work.

II. State of the art

A. Simulation platforms

There is a large diversity of traffic simulators based on multi-agents systems or cellular automata. However, most of these applications are commercially oriented and exposes only very few design details. We can especially name Synchro Studio[1] with SimTraffic, Aimsun[2], PTV Vissim[3], Paramics[4] or also TransModeler[5].

On the fringes of those systems we also find corporate software like ArchiSim [1], SCANeR II [2] or Megaffic [5]

[1] http://www.trafficware.com/products/planninganalysis-software
[2] http://www.aimsun.com/
[3] http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/
[4] http://www.paramics-online.com/
[5] http://www.caliper.com/transmodeler/

which are based on the multi-agent paradigm [6], [7]. Nevertheless, these tools are dedicated to specific issues of IFSTTAR (the French research institute on transport), Renault and IBM, such as targeted studies about drivers psychology, the vehicle cabin ergonomics or else massive microscopic simulations. Their architectures are custom built (in terms of infrastructure software and hardware), behaviors are specialized and hardly extensible. Similarly, the DIVAs [8] framework (initiative of the University of Texas at Dallas) is more centered on agent vision problems.

Finally, some open source alternatives exist, such as MATSim [9], SUMO [10] or MITSIMLab [11]. Only MATSim and SUMO are based on a multi-agent system. Both allow the usage of geographical data (GIS). They also have the advantage to simulate a large number of vehicles, at the price of a poor flexibility of the simulation mechanisms. Vehicles own only a small number of behaviors, reducible to a car-following model and the lane changing feature. Such simple and naive behaviors are also provided in the multipurpose platform GAMA [12]. In all cases, the revision and extension of the behavioral model is the main issue.

B. *Behavioral models*

In most traffic simulators, modeling efforts essentially focus on vehicles which are in general *the only agents* in the system. All the traffic complexity is modeled through the perception, cognition, actions capabilities of those agents only and through their behaviors and interactions. Morevover, the majority of modeling works try to address issues in multiple areas: perception (e.g. active perception [13], virtual lanes [14], vision cones [15]), drivers psychology [16], their report to standards [17], [18] or also the usage of game theory to manage conflicts in crossroads [16].

Though the majority of these models have been subject to separate validations, the combinations of their underlying assumptions obviously raise serious evaluation issues. An accurate psychological model of drivers does not guarantee indeed that the population of simulated vehicles is more realistic at the macroscopic level. To design a tool which allows to test and compare various scenarios or assumptions, the problem is not to choose and implement one particular psychological theory or another one, but rather to permit the user to:

1) access *explicitly* (and intelligibly) the behavioral models that reflect these assumptions,
2) choose those to use in a scenario,
3) redesign easily the experiment, without having to rewrite any line of code.

Before explaining how we implement this approach in *TrafficGen*, we have to present the issue of retrieving information for the simulation.

III. Open format data

For some time, the increasing number of initiatives aimed at providing geographical data in different open formats, provides an opportunity for, on the one hand, building on-the-fly, up-to-date multi-agent environments for traffic simulation and, on the other hand, reproducing real-time traffic state in a simulation (or compare the outputs of a particular hypothesis to the real situation). We present in this section some of these formats and their usage.

A. *Cartographic data*

During the last fifteen years there has been a lot of new cartographic and routing online services. Among them, the *OpenStreetMap* project (OSM[6]), created in 2004 at the University College, London, is a participatory initiative based on users contributions (like Wikipedia). Elements listed in *OpenStreetMap* are extremely diversified (details and functions of building, speed limits, tourist information, etc.) and the data format is open.

We can also mention *OpenDrive*[7], which is an open format developed by private companies in 2005 (VIRES Simulationstechnologie GmbH). This format focuses more specifically on road features (geometry of the pavement, slopes, elevation, nature of the coating, etc.) and was developed in order to standardize the simulation of vehicle prototypes on test circuits.

B. *Flow information*

Besides this cartographic information, a simulation must be able to take into account realistic vehicles flows. Though it is possible to use flow generators (e.g. MNTG [19]) which generate vehicles with scripted routes from a map, real traffic data is obviously a better way either to calibrate the flows of simulated vehicles or to validate simulation ouputs. The availability of such data is currently increasing both at national and urban scales.

For instance the French National Traffic Information Centre ("*Bison Futé*") provides freely and in real time (every 6 min.) data about the main highways of the national road network and the main cities[8]. Those data use the Datex II format[9] (i.e. European exchange format for road traffic information), which indicates events (car crash, traffic jam, etc.) but also traffic indicators for each measure point of the network: flow rate (number of vehicles per period of measure), occupancy rate (ratio between total vehicles length on a section and length of this section) and mean vehicles speed. These informations allow to recreate a population of simulated vehicles, which owns similar characteristics.

[6]http://wwww.openstreetmap.org

[7]http://www.opendrive.org

[8]http://diffusion-numerique.info-routiere.gouv.fr

[9]http://www.datex2.eu

Likewise, most of big cities have static measurement systems: e.g. in France, SIREDO[10] based on magnetic loops under the road, which provides information every 15 min (flow, mean speed, etc.) or each hour (occupancy rate). In addition, punctual measurements with pneumatic tubes may often complement this information.

C. Road infrastructure creation

A road can be represented as a sequence of nodes knowing their possible predecessors or successors. Vehicles move along a straight line from a node to the next one and get the next possibles routes when they arrive on a node. In *TrafficGen*, vehicles have a logical position on the axial line of the road: the visualization of the road or the position of vehicles on his lane are only a graphic representation as shown on figure 1.

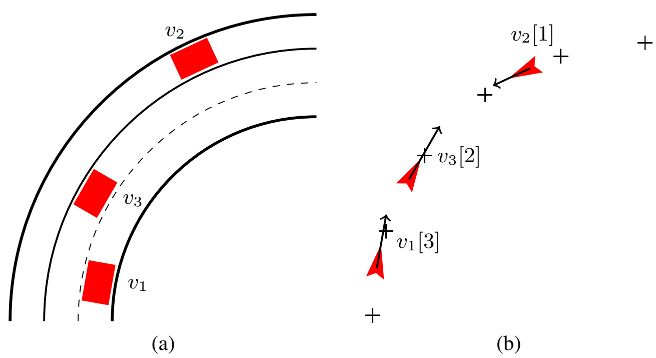


Figure 1. (a) Road section with three lanes having one vehicle per lane. (b) *TrafficGen* representation: crosses are nodes, each arrow is the speed vector of a vehicle agent, the number between brackets is the vehicle lane.

Thus, the initialization of the simulation environment involves to read nodes and ways from a database, so as to instantiate them and build the graph defining the road network topology. Yet, some nodes are just an inflection point in the road (only one possible output), while others are more complex junctions which bring several directions (i.e. crossroads) and imply that the vehicles make a choice. Moreover, GIS data (i.e. OpenStreetMap or OpenDRIVE) provide indications on road infrastructure which have to be reified in the simulation (e.g. traffic signals, stops, etc.) to enhance the realism of vehicles behaviors. More generally, the capability of integrating other road network elements (e.g. pedestrian, cycles, bus lanes, tram lines, etc.) implies to associate GIS elements with corresponding entities situated in the environment. Finally, in order to test scenarios the simulation needs to include vehicles generation points (able to reproduce flows from real data) and measurement tools.

Considering the large diversity of these entities and the need to extend them according to modeling issues and goals, *we advocate a clear separation between declarative and procedural model aspects, but also an approach where all entities are handled homogeneously*. Therefore, we use the interaction-oriented approach "IODA" [20], [21] which states that every entity is an agent and that every behavior must be defined as a generic rule between agents (called an *interaction*). This leads to the design of separate, reusable agents and interactions libraries, which are processed by a generic simulation engine. Concretely, the agents used in *TrafficGen* are described below (we present the modeling of their behavior in the next section). A dictionary ensures the correspondence between GIS elements and agents.

Nodes have other nodes as acquaintances; they are connected by **links** representing ways, endowed themselves with attributes (e.g. number of lanes, one-way or two-way, speed limits, etc.).

Vehicles have a desired speed (which they try to reach) and an instant speed they adapt depending on the situation. They also compute a mobile average speed (over the last simulation cycles). As shown on figure 1, all vehicles move along the road axis (i.e. a link between two nodes) and are logically affected to a lane. The vehicle perception is customizable; by default each vehicle has a vision cone in the front and another one in the back. At each node, it chooses a destination node and memorizes where it comes from. It is also possible to specialize vehicles into "subspecies" with different perception or action capabilities (e.g. cars, cycles...).

Crossroads are created when a node owns more than two acquaintances. They manage the access to lanes to avoid or detect collisions. Different possibilities have been proposed to do this [16], [22]; as a rough default behavior, we use a semaphore to control the access to each lane. Yet, in the IODA approach the crossroad agent does not own the regulatory mechanisms: instead, they are expressed through rules (*interactions*), so that different mechanisms from the literature can be implemented as well and assigned to the crossroads afterwards.

Traffic signals and others road signs are created from GIS information (when available) and reified into agents belonging to a specific family.

Generators are in charge of vehicles creation at a rate, and with characteristics, which reflect either real data or probability laws.

Probes measure and save characteristics of passing vehicles. They can be customized so as to inspect features that are considered relevant in each simulation scenario.

Event managers of several kinds allow to extend the actions that can be performed over vehicles or other agents according to specific scenarios; especially, **speed reducers** can impose temporary speed limits to all vehicles within a circular perimeter.

The last three agents families are obviously not part of GIS data. Generators and probes can be placed on real

[10]for *Système Informatisé de REcueil de DOnnées*: http://www.transport-intelligent.net/produits-services/article/siredo

measurement points (to generate realistic vehicles flows or compare the simulated flows with real data) or on arbitrary nodes for experimentation. Event managers can be put everywhere without node constraints: they can be used to evaluate the impact of emergency measures, such as the local speed reduction policy in response to pollution peeks for instance.

IV. Behavioral model

Traffic simulations involve complex vehicle behaviors. Part of this complexity, especially everything in relation with the way drivers get information in their environment, can nevertheless be delegated to agents which reify road devices (crossroads, traffic signs, lights, etc.), as suggested by Gibson in his *affordance theory* [23]. Indeed, our goal is not to reproduce a psychologically realistic model of the driver, but to ensure that vehicles move in a manner consistent with knowledge and observations, and yet to maintain their diversity.

Aiming at developing tools to help decision, it is important to allow the user to build scenarios, revise them easily, compare them and evaluate their outcomes. For example: *It is 10 am at Paris, 80% of drivers are elderly person, 20% are dynamic executives and arrive on the ring road at a rate of 50 vehicles/min. A pollution alert is triggered on Paris inner centre. What would be the result on the traffic density in main avenues entries of a speed limitation to 50 km/h on the ring road ?*

To take into account so many aspects (the subnetwork to study, the individual specifications of vehicles, their input flow), the simulator architecture must allow modular associations of various agents families to different generic behaviors.

Our proposition is to use the matrix representation of the IODA approach [21] wherein we indicate which interactions can be performed between each pair of families of agents (Table I). For example, when a vehicle A (*source*) perceives a vehicle B (*target*), it may *overtake*, *decelerate* or execute an *emergency braking* depending on their speeds and the distance between them.

Table I

Simplified extract of the interaction matrix describing the behavioral model in *TrafficGen*. It expresses which interactions can be performed by some agents on others, depending on their family: e.g. *cars* (kind of *vehicles*) can *Overtake*, *SlowDown* or *EmergencyBrake* on other *cars*.

Sources \ Targets	∅	Cars	Nodes	Crossroads	…
Cars	Accelerate Forward Fallback	**Overtake** **SlowDown** **EmergencyBrake**	Cross	Enter Exit	…
Nodes					…
Crossroads		SelectDirection			…
Lights		Stop			…
Generators	CreateVehicle				…
Probes		Count			…
⋮	⋮	⋮	⋮	⋮	

Table II

"Naive" algorithm for action selection by a car, based upon the perceived characteristics of a neighboring car (*target*)

```
if target:distance ≤ 10 ∧ not-in-crossroad? ∧ target:in-front? ∧
target:same-direction? ∧ target:same-lane? ∧ not-road-end? then
    // Overtake(30; 10)
    if target:too-close? ∧ target:too-slow? ∧ has-left-lane? ∧ left-
    lane-free? then
        save-target-overtaking
        go-to-left
        forward
    // EmergencyBrake(20; 5)
    else if target:distance ≤ 5 ∧ target:emergency-distance? then
        emergency-braking
    // DecelerateAvoidCar(10; 10)
    else if target:too-close? ∧ not-stopped? then
        slowdown
        forward
    end if
end if
```

If we write a "naive" algorithm (Table II) for the decision-making between these three behaviors, we observe the interweaving of several distinct concepts. It brings together the definition of actions to perform with a complex decision process. We can also observe that the notion of priority between behaviors is not explicitly specified. This causes a lack of modularity in the behaviors. In fact, a mere attempt to modify the priority of an interaction requires to change the order of the "if" cascades so as to take into account execution conditions and the distance between agents.

On the contrary, within the IODA approach, our example leads to define the appropriate associations in the interaction matrix (Table III) and three independent interactions (Table IV). This representation makes a clear separation between the definition of actions and their conditions through *interactions*. It also allows an explicit scheduling of behaviors (through priorities) and a "distance guard" made explicit in the interaction matrix.

These interactions specify behaviors as conditions/actions rules (like in STRIPS) by using abstract primitives (Table IV). These primitives can lead to a different implementation in each agent family, depending on its structure and capabilities. Thanks to this approach, a revision of the model almost consists in modifying the interactions assigned to agents families in the matrix, together with their priorities and their distance guards.

This approach facilitates the revision of the models. Moreover, the separation of knowledge relative to entities and their behavior, allows to change the model even during the simulation simply by modifying the interaction matrix. Furthermore, interactions (as generic rules) can be reused in various contexts.

Table III
REPRESENTATION OF AVAILABLE CARS/CARS INTERACTIONS OF TABLE I IN IODA-NETLOGO. THE FIELDS REPRESENT THE FOLLOWING ITEMS RESPECTIVELY: THE TYPE OF **SOURCE** AGENT (E.G. `cars`); THE NAME OF THE **INTERACTION** (E.G. `Overtake`); THE **PRIORITY** OF THE INTERACTION (E.G. `30`); THE TYPE OF THE **TARGET** AGENT (E.G. `cars`); THE **DISTANCE GUARD** (E.G. `10`)

cars	Overtake	30	cars	10
cars	EmergencyBrake	20	cars	5
cars	DecelerateAvoidCar	10	cars	10

Table IV
IMPLEMENTATION OF THE INTERACTIONS DEFINED IN TABLE III. THE **TRIGGER** DESCRIBES MOTIVATIONS TO PERFORM THE INTERACTION, THE **CONDITION** EXPRESSES PREREQUISITES OF ACTIONS AND THE **ACTIONS** ARE RUN IN SEQUENCE.

```
INTERACTION Overtake
    TRIGGER target:in-front? target:same-direction?
            target:same-lane? target:too-close?
            target:too-slow?
    CONDITION not-in-crossroad? not-road-end?
              has-left-lane? left-lane-free?
    ACTIONS save-target-overtaking
            go-to-left
            forward
END

INTERACTION EmergencyBrake
    TRIGGER target:in-front? target:same-direction?
            target:same-lane? target:emergency-distance?
    CONDITION not-in-crossroad? not-road-end?
    ACTIONS emergency-braking
END

INTERACTION DecelerateAvoidCar
    TRIGGER in-front? target:same-direction?
            target:same-lane? target:too-close?
    CONDITION not-in-crossroad? not-stopped? not-road-end?
    ACTIONS decelerate
            forward
END
```

V. THE *TrafficGen* PLATFORM

As a proof of concept, we have developed an experimental tool named *TrafficGen*, based on the IODA extension[11] for the NetLogo multi-agent platform. NetLogo [24] also provides a native GIS extension and a third-party SQL extension[12]. We first tested the capacity of this platform to import maps from various contexts (e.g. figures 2, 6 and 7), then implemented "basic" behaviors like collision-free and overtaking models (figure 7).

A. *OpenStreetMap*

In this part, we focus on the OSM format, explaining how we build a road network in simulation from OSM data. Our approach can though easily be transposed to other transport networks as shown in section V-B.

OSM Data are structured as an XML file composed of three levels of elements. **Nodes** (`<node>`) are points of interest on the map, identified by an ID and GPS coordinates. Like all OSM elements they can encapsulate key/value **tags** (`<tag k="..." v="..." />`) giving complementary information. **Ways** (`<way>`) are nodes sequences (`<nd ref="..." />`) which can represent as well an open curve (e.g. a street) or a polygon. Ways also own tags (e.g. one-way road, number of lanes, roundabout, etc.). **Relations** (`<relation>`) are logical entities involving nodes and ways (`<member type="node|way" ref="..." role="..." />`) with particular roles (e.g. subway lines are made of ways which connect nodes, and of special nodes which are the stations). Each relation can also be endowed with specific tags.

[11] http://www.lifl.fr/SMAC/projects/ioda/ioda_for_netlogo/
[12] https://code.google.com/p/netlogo-sql/

As we can see, this information is low-level and loosely structured. Yet, the tags system allows to associate various information to an element, such as: speed limitation, cycles lane, public lights, etc. But in the case of road networks the most important tag is keyed by `highway`; the associated value indicates the nature of the corresponding element (i.e. the type of road: `primary`, `secondary`, `residential`...).

To build a road network from an OSM file, the data have to be filtered to keep only relevant information about the network structure. Lots of tools can be used (e.g. *Osmosis*[13], *JOSM*[14], *Merkaartor*[15]) to maintain or exclude nodes and ways depending on their tags (i.e. here we are looking for *highway* tags). Then, we recommend to inject data into a "true" database, to facilitate access to OSM elements properties (i.e. tags values). PostgreSQL and the PostGIS

[13] http://wiki.openstreetmap.org/wiki/Osmosis
[14] http://wiki.openstreetmap.org/wiki/Josm
[15] http://wiki.openstreetmap.org/wiki/Merkaartor

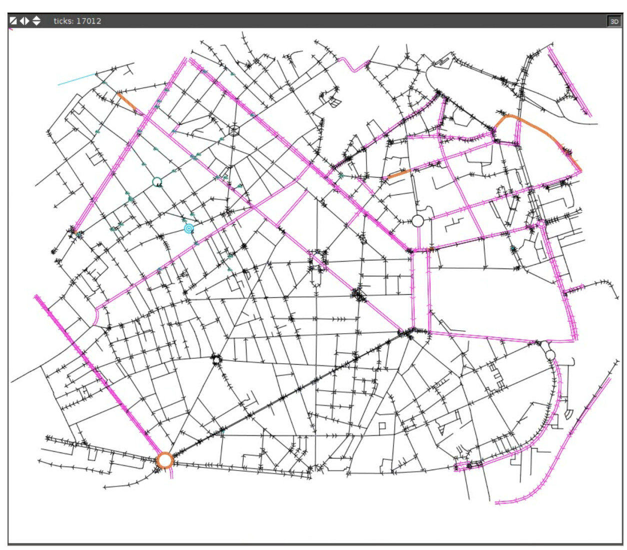


Figure 2. Part of the city center of Lille (France) loaded in *TrafficGen*.

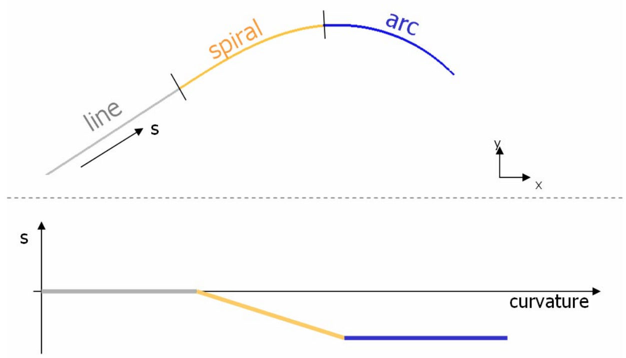


Figure 3. Extract of the OpenDRIVE specification illustrating the different geometries. The curve above shows the geometrical aspect of each geometry according to the curvature coefficient (graph below).

extension are particularly adapted to handle this. A tool like *osm2pgsql*[16] makes this operation in one command line.

B. OpenDRIVE and other transport networks

In order to demonstrate the strength of our solution facing different data formats, we applied it also to the OpenDRIVE format. The OpenDRIVE format is composed of three distinct parts: roads, controllers (i.e. traffic lights and dynamic speed limitations) and junctions (e.g. crossroad, roundabout, etc.), which are represented by the `road`, `controller` and `junctions` markers. Unfortunately, there is no simple way to parse the OpenDRIVE format, so we have developed a dedicated plug-in (not described here) to load the data into our simulator.

For now we have focused on roads implementation. Roads are composed of three layers, which allow different detail levels. Here we only address the first layer (the whole specification can be found on the OpenDRIVE website[17]), which is dedicated to road geometry. The layer is composed of a set of roads (i.e. `road` markers), each one owning a list of sections with a specific geometry (i.e. `geometry` markers). In OpenDRIVE 1.3 four types of geometries can be use: straight lines (`line`), curves (`arc`), spirals (`spiral`) and third degree polynomials (`poly3`). The differences are mainly in the curvature coefficient which are respectively zero, nonzero constant and linear, as shown on figure 3.

In this format, each geometry only has one coordinate (usually a starting point), a length and information about the curvature. Thus, nodes agents must be extrapolated according to the geometries. Nodes are then connected to build ways (Figure 4). Noteworthy, this format has a major advantage over a solution like OSM, in the point of view of multi-scale simulations [25], since the interpolation level can be tuned to the desired scale.

[16]http://wiki.openstreetmap.org/wiki/Osm2pgsql
[17]http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1.3D.pdf

We have also applied our method to other transport networks. For example, we used OpenStreetMap to get data from various french subway networks (e.g. subway of Lyon on figure 5). In order to complete the network creation, required agents have been mapped to OSM elements (i.e. trains, stations, etc.), additional interactions have been defined (possibly by using existing interaction primitives), and the corresponding interaction matrix has been written.

C. Experimental example

To illustrate our approach, we show a simple experiment based on the map of the city centre of Lille (figure 2) wherein a measure point gives the travel time (relative to an upstream point) and density (i.e. number of vehicles per period of 100 simulation cycles) of observed vehicles (figure 8). These are created by a *generator* agent with a random interval following a Poisson law (with parameter $\lambda = 5$ cycles) and a speed dictated by a normal law (with a mean speed of 90 km/h and a standard deviation of 10 km/h).

At simulation cycle 1500, a speed limitation to 50 km/h is imposed by a speed reducer agent in a perimeter of 800 m around the measurement point. As we can see on the figure 8a, the limitation increases progressively the travel time of vehicles, but this one does not stabilize, which indicates the formation of a traffic jam. The density plot (figure 8b) confirms this observation: after a small diminution (due to the slowing of cars before the measure point) the density increases considerably.

To realize this kind of experimentation, we only have to place a generator, a probe, and an event manager in the study environment. The behaviors than we have implemented in *TrafficGen* are profusely cited in the literature and most of them are already experimentally validated. Thus we mainly have to validate the integration of data flow (ongoing work) and the reproduction of an agent population from these global information. We plan to join national information (e.g. *Bison Futé*) with information from the urban communities, by using techniques developed by Champion et al. [16] or Lacroix et al. [3].

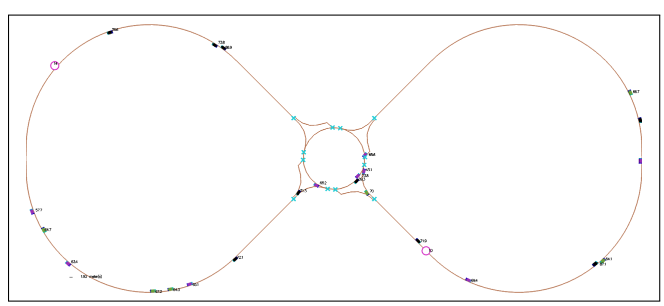


Figure 4. An OpenDRIVE file loaded in *TrafficGen*

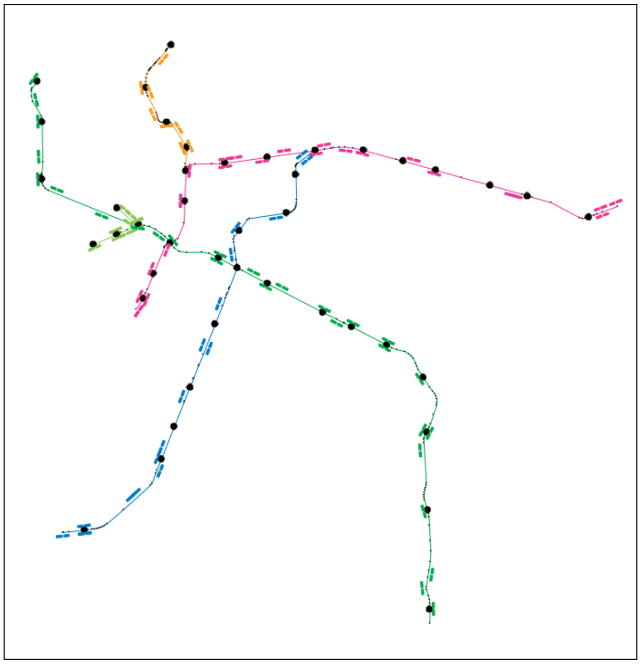


Figure 5. A simulation of the subway of Lyon (France) in *TrafficGen*.

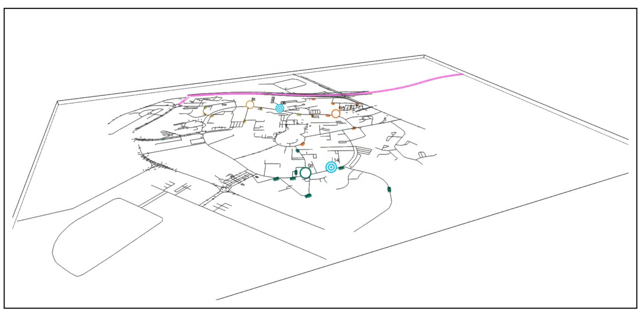


Figure 6. The Lille 1 University campus in *TrafficGen*.

VI. Conclusion and perspectives

Simulators are currently recognized as essential tools for decision making. Transport in general, and traffic road in particular, are more and more studied, both at a macroscopic level (flow equations, aggregate variables, etc.) and at the microscopic level (individual-based approach). These last years the amount of online real data increased constantly. In this article we have proposed a modular and highly tunable integration method of these data into an intelligible, easily revisable behavioral model for multiagent simulation. In addition, our approach is extensible to other transport networks (e.g. trams, bus, subways, cycles, pedestrians, etc.) with little coding effort, and to other data formats such as OpenDRIVE.

Though the quality of open data is questionable [26], their usage facilitates the realization of a traffic simulation by providing free data which are continuously updated. The IODA approach pushes the designer to elaborate behavioral

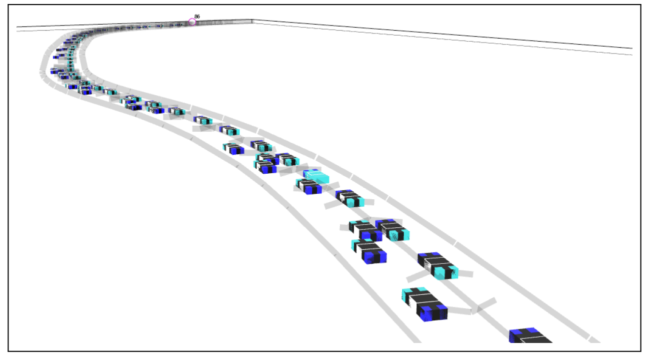
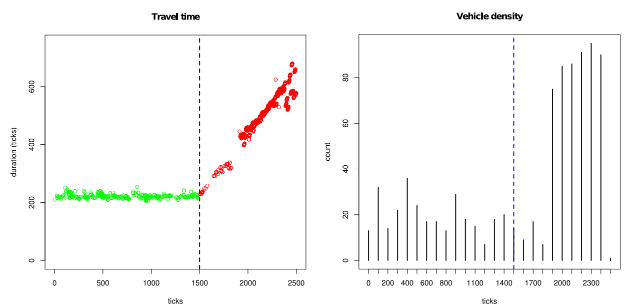


Figure 7. Traffic on French highway A23 in *TrafficGen*.



(a) (b)

Figure 8. Evolution of travel time (a) and density (b) (number of vehicles per 100 ticks) of observed vehicles. At simulation cycle 1500 a speed limitation is placed in a radius of 800 m around the measure point, creating a traffic jam.

models in an incremental process, separating the structure and ability of agents from abstract behavioral rules, which make the models easy to extend or change. All elements of the road infrastructure likely to participate in the behavior of the system are in fact modeled and implemented by agents, and the behaviors of agents are based on generic rules (the interactions).

These properties allow a large flexibility in the achievement and testing of scenarios which can cover a wide range of characteristics, from the road network choice to the study of agents behavior details.

Our ongoing work focuses on the acquisition of flow information in these various networks, which are most of the time less standardized than those of road networks and are often provided as schedule grids. Nevertheless, the generalization of our approach should facilitate the design of simulation “on demand” to answer to relatively various scenarios.

References

[1] S. Espié, F. Saad, B. Schnetzler, F. Bourlier, and N. Djemane, “Microscopic traffic simulation and driver behaviour modelling: the ARCHISIM project,” in *Conf. Road Safety in Europe and Strategic Highway Research Program (SHRP)*, 1994, pp. 22–31.

[2] A. Champion, R. Mandiau, C. Kolski, A. Heidet, and A. Kemeny, “Traffic generation with the SCANeR™ simulator: towards a multi-agent architecture,” in *Driving Simulation Conference*, 1999, pp. 311–324.

[3] B. Lacroix, P. Mathieu, and A. Kemeny, “Formalizing the construction of populations in multi-agent simulations,” *J. Eng. App. of AI*, vol. 26, no. 1, pp. 211–226, 2013.

[4] P. Mathieu and S. Picault, “From real purchase to realistic populations of simulated customers,” in *11th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, ser. LNCS, Y. Demazeau *et al.*, Eds., vol. 7879. Springer, 2013, pp. 216–227.

[5] T. Osogami, T. Imamichi, H. Mizuta, T. Morimura, R. Raymond, T. Suzumura, R. Takahashi, and T. Idé, “IBM Mega Traffic Simulator,” IBM, Tech. Rep., 2012.

[6] B. Chen and H. Cheng, “A review of the applications of agent technology in traffic and transportation systems,” *Trans. Intell. Transport. Sys.*, vol. 11, no. 2, pp. 485–497, 2010.

[7] A. Bazzan and F. Klügl, “A review on agent-based technology for traffic and transportation,” *The Knowledge Engineering Review*, vol. FirstView, pp. 1–29, 1 2014.

[8] M. Al-Zinati, F. Araujo, D. Kuiper, J. Valente, and R. Wenkstern, “DIVAs 4.0: A multi-agent based simulation framework,” in *IEEE/ACM 17th Int. Symp. on Distributed Simulation and Real Time Applications*, 2013, pp. 105–114.

[9] B. Raney and K. Nagel, “An improved framework for large-scale multi-agent simulations of travel behavior,” *Towards better performing European Transportation Systems*, pp. 305–347, 2006.

[10] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO - Simulation of Urban MObility,” *Int. J. on Advances in Systems and Measurements*, vol. 5, no. 3–4, pp. 128–138, December 2012.

[11] Q. Yang, “Simulation laboratory for evaluating dynamic traffic management systems,” Thèse de doctorat, Massachusetts Institute of Technology, 1997.

[12] P. Taillandier, D.-A. Vo, E. Amouroux, and A. Drogoul, “GAMA: A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control,” in *Principles and Practice of Multi-Agent Systems*, ser. LNCS, N. Desai *et al.*, Eds., vol. 7057. Springer, 2012, pp. 242–258.

[13] L. Bourgois, J. Saunier, and J.-M. Auberlet, “Towards contextual goal-oriented perception for pedestrian simulation,” in *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART)*, F. J. and A. Fred, Eds. SciTePress, 2012, pp. 197–202.

[14] L. Bonte, S. Espié, and P. Mathieu, “Virtual lanes interest for motorcycles simulation,” in *5th European Workshop on Multi-Agent Systems (EUMAS’07)*. ATIA, 2007, pp. 580–596.

[15] D. Kuiper and R. Wenkstern, “Agent vision in multi-agent based simulation systems,” *J. of Auton. Agents and Multi-Agent Systems*, pp. 1–31, 2014.

[16] R. Mandiau, A. Champion, J.-M. Auberlet, S. Espié, and C. Kolski, “Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation,” *Appl. Intell.*, vol. 28, no. 2, pp. 121–138, 2008.

[17] B. Lacroix, P. Mathieu, and A. Kemeny, “Generating various and consistent behaviors in simulations,” in *Proceedings of the 7th International conference on Practical Applications of Agents and Multi-Agents Systems (PAAMS’2009)*, ser. Practical Advances in Intelligent and Soft Computing, Y. Demazeau *et al.*, Eds., vol. 55. Springer, 2009, pp. 110–119.

[18] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espié, “A behavioral multi-agent model for road traffic simulation,” *J. Eng. Appl. of AI*, pp. 1443–1454, 2008.

[19] M. Mokbel, L. Alarabi, J. Bao, A. Eldawy, A. Magdy, M. Sarwat, E. Waytas, and S. Yackel, “MNTG: An extensible web-based traffic generator,” in *13th Int. Symp. on Spatial and Temporal Databases (SSTD)*, ser. LNCS, vol. 8098. Springer, 2013, pp. 38–55.

[20] Y. Kubera, P. Mathieu, and S. Picault, “Everything can be agent!” in *9th Int. Joint Conf. on Auton. Agents and Multi-Agent Systems (AAMAS)*, W. van der Hoek *et al.*, Eds. IFAAMAS, 2010, pp. 1547–1548.

[21] ——, “IODA: An interaction-oriented approach for multi-agent based simulations,” *J. of Auton. Agents and Multi-Agent Systems*, vol. 23, no. 3, pp. 303–343, 2011.

[22] M. Tlig, O. Buffet, and O. Simonin, “Cooperative behaviors for the self-regulation of autonomous vehicles in space sharing conflicts,” in *IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2012, pp. 1126–1132.

[23] J. Gibson, *The Ecological Approach to Visual Perception*. Hillsdale, 1979.

[24] U. Wilensky, “Netlogo,” 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/

[25] S. Picault and P. Mathieu, “An interaction-oriented model for multi-scale simulation,” in *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI’2011)*, T. Walsh, Ed. IJCAI/AAAI, 2011, pp. 332–337.

[26] J.-F. Girres and G. Touya, “Quality assessment of the french OpenStreetMap dataset,” *Transactions in GIS*, vol. 14, no. 4, pp. 435–459, 2010.