Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 56 (2015) 168 – 176

Procedia
Computer Science

The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015)

Towards the Dynamic Evaluation of a Public Bus Network for Small Size Urban Environments

Olivier Lamotte[a,*], Nicolas Gaud[a], Stéphane Galland[a]

[a]IRTES, Université de Technologie de Belfort-Montbliard, Rue Ernest Thierry-Mieg 90010 Belfort, France

Abstract

In this paper, we present models and the corresponding decision-support tool, SARTRE, for evaluating the Public Transport Networks (PTN), and more specifically a bus network. Firstly, we define the framework architecture of SARTRE. The PTN model is explained, and various evaluation and assessment methods are explained. These models and evaluation algorithms are specifically designed to address to the design and the supervision of PTN for small- and medium-size towns. This approach has been successfully implemented in a decision-support tool, which have been used by the public transport authority of the region of Belfort in France to supervise its new PTN.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Public Transport System, Data-Storage System, Evaluation Indicator, Bus Network, Small-Size City

1. Introduction

Public Transport Network (PTN) Design[1] is a scientific and technological domain that aims at designing the best public transport services for a specific region. In this paper, the PTN Design in the context of small- and medium-size towns (less than 100,000 people) is considered. Indeed, these towns are too small to use the same methods and tools than bigger cities, because the deployment of such methods is too costly in terms of time, financial resources and technical expertise. Moreover, they are more sensible to the private operating costs, the social costs of the restructured system, and the investment costs for the bus fleet and the road infrastructures. Unfortunately traditional tools for PTN evaluation and simulation have several drawbacks in the context of small- and medium-size towns:

- Accessibility: specific and high competencies and skills are required to use these tools;
- Efficiency: it is still long and difficult to create bus network models;
- Understandability: it is difficult to understand outputs provided by the existing tools without the help of (expensive) experts of the domain;

* Corresponding author. Tel.: +33 384 583 344; fax: +33 384 583 342.
E-mail address: olivier.lamotte@utbm.fr

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.07.191

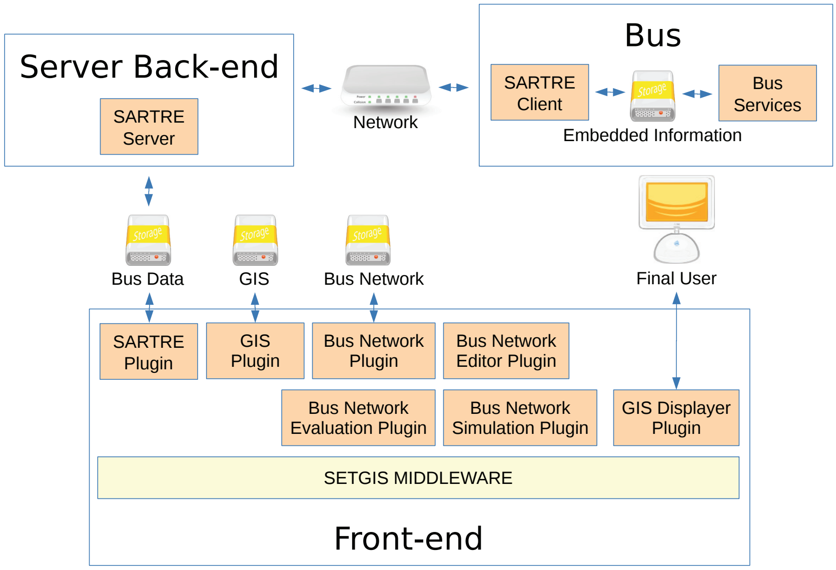

Fig. 1. Architecture of the SARTRE Framework

- Scalability: they are designed for large towns, not for small- and medium-size towns.

SARTRE is a decision-support tool implementing the various models and the algorithms for the evaluation of small- and medium-size PTN. It provides models, features, and tools to solve the problems described above. It (i) **collects data from embedded sensors and devices in buses**; (ii) **stores collected data into a central database**; (iii) imports and edits the data collection from Geographical Information Systems (GIS), e.g. shape files; (iv) imports and displays geo-referenced pictures; (v) import and edits the definition of the bus network; (vi) **computes indicators for evaluating the bus network quality**. In this paper, the **works in progress** related to the design of an architecture, and the determination of the evaluation indicators are presented.

The paper is structured as follows. Section 2 presents the overall architecture of the SARTRE model and the embedded services. Section 2.1 describes the major services that are embedded in the buses. Section 3 gives the first evaluation indicators that are provided by the SARTRE model. Section 4 briefly discusses several related works. And, Section 5 concludes this paper and describes the next steps for finalizing the works presented in this paper.

2. Architecture of the SARTRE Framework

The SARTRE framework contains a Digital Data Storage (DDS) for collecting and storing the data from the buses. Figure 1 illustrates the global architecture of the SARTRE framework. Three majors components are proposed: (i) the front-end application, (ii) the back-end server, and (iii) the embedded services in the buses.

The front-end application provides to the final user the graphical user interface for extracting and displaying the data in the three main database of the SARTRE framework.

The geographical information database (GIS) contains the definition of the road network. Each road segment is linked to other road segments through road connection: cross road, etc. Road segments and connections are gathering into a common structure: the road network. The left side of Figure 2 presents the class diagram of the main concepts of the GIS database.

The bus network database contains the definition of the bus lines (illustrated by the right side of Figure 2). The finest primitive in a bus network is the bus stop. A bus stop is a place where buses stop allowing passengers to board or get off the bus. Bus stops are normally positioned on the roadside and are distinct from facilities such as bus stations. The construction of bus stops tends to reflect the level of usage. Bus itinerary is a unidirectional bus route from a

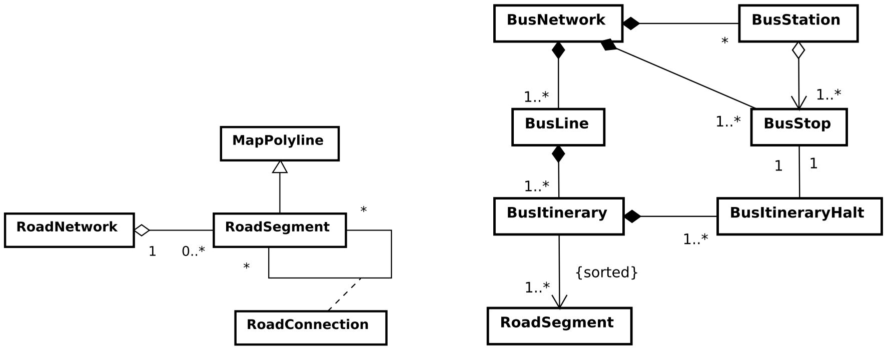

Fig. 2. UML Class Diagrams of the Road and Bus Networks

terminal to another, and containing an ordered list of bus stops. Bus line is a collection of bus itineraries with same name or label.

The back-end server provides a centralized database that contains the data collected inside the buses. This database is synchronized with the databases that are embedded inside the buses (Embedded Information). The data are collected by the bus services that are described in Section 2.1.

Figure 3 focuses on the architecture of the part of the framework that runs inside the buses. The major embedded services are described in Section 2.1. The services related to the synchronization of the bus database, and the central bus database are described in Section 2.2.

2.1. Specific Embedded Services

In this section, five software services that are embedded inside the buses are presented. Figure 3 illustrates the relationship between these services with usage links.

2.1.1. Area Detection Service

This embedded service is a building block that supports other higher-level services. Like every embedded service of our architecture, it follows the design pattern Observer to warn each of its listeners when the bus entered or exited a particular area. These events allow the other services to implement specific operations that are related to the place.

To operate, this service listens the GPS device that is embedded in the bus, and makes comparisons with a list of areas of interest. Each area is constituted of a geographic localization and a specific radius. Every time that the bus position is updated, a belonging test is performed for determining the areas on which the bus is located. For each region of the list, one of the four following events is thrown:

- `Entered`, if the bus entered in the area;
- `Exited`, if the bus exited of the area;
- `OutOfScope`, if the bus is out of the area scope; and
- `EnteredAndExited`, if the bus is entered and exited of the area, according to the speed of the bus and the timestamp between each GPS update.

As the accuracy of GPS data is not constant, the events are not sent immediately. Rather, the emission is performed after a specified number of updates of the location data. This mechanism avoids sending multiple events when the bus is approaching an area.

2.1.2. Map Matching Service

This service aims to locate the collected data by relating the GPS coordinates to the definition of the bus network. This operating mode allows to perform queries more efficiently, based on the network topology. For example, it is

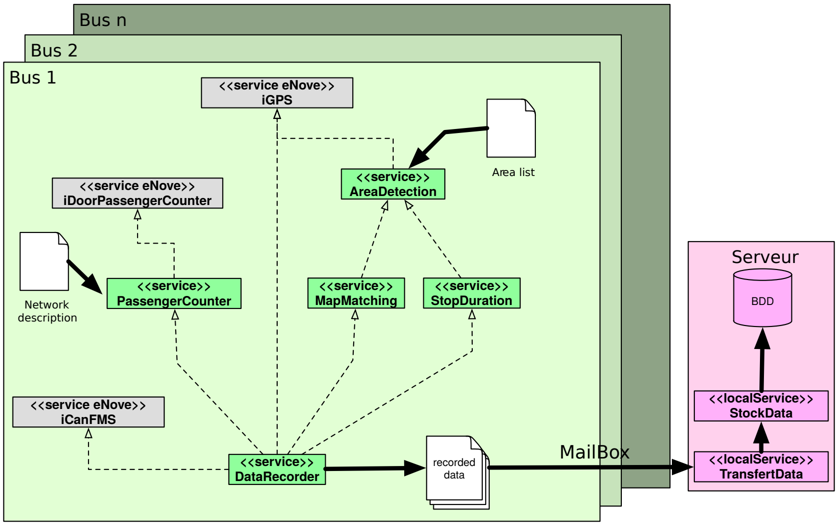

Fig. 3. Architecture of the Embedded Framework

possible to recover all the instantaneous speeds of the buses running in a bus itinerary of a particular line. Without this mechanism, this query would be complex and inefficient due to the need to isolate the records related to the GPS positions and the bus line in a joint virtual table.

For tagging each record with the line and route information, the map matching service uses a graph representing the bus network, and a set of locations representing each bus stop. The graph contains the routes followed by the buses. When two lines go into the same street, they are merged and represented by a single segment in the graph. According to the model presented on Figure 2, each definition of a bus itinerary contains the sequence of bus stops, either terminus or standard bus stop.

The map matching algorithm uses these data structures for retaining at which bus stops the bus was detected. Since, the bus itinerary that is followed by the real bus is not known a priori by the software system[1], the algorithm keeps in its internal data structure the set of bus stops that was encountered after matching the GPS location to the graph of road segments. This set of bus stops permits to constitute the set of candidates for the bus itinerary that is followed by the bus. More the algorithm detects new bus stops; more bus itineraries could be removed from this set.

In the same way as for the other embedded services, the map matching service sends events only when a bus stop is crossed, and when the itinerary is determined. The listening services retrieve the ordered list of bus stops crossed by the real bus with the corresponding date of passage, and the bus itinerary.

2.1.3. Stop Duration Service

One of the most important data for the analysis of the bus network performance is the time spent by each bus at stops. Indeed, buses are scheduled with a high frequency (about 10 minutes between each bus), and each delay caused by a prolonged stop at a bus stop could penalize the operation of the entire network.

For enabling a detailed analysis of the network behavior, a service that is responsible for recording data on passengers boarding/get-off stage is integrated in our system. Based on an event mechanism, the service throws a stamped event when: (i) the bus enters into a given area; (ii) the bus has stopped; (iii) the first door is opened; (iv) the last door is closed; (v) the bus circulates; and (vi) the bus exists from the area.

[1] The embedded software system knows the bus line, but not the bus itinerary. This line information is used for updating the information messages on the embedded screens.

This mechanism enables the determination of the time spent by a bus for a specific bus stop. After analyzing these data over a period of time, the experts are able to explain observed dysfunctions of the bus network, and adjust the time tables of the buses.

2.1.4. Passenger Counting Service

In direct connection with the previous service, the passenger counting service enables the computation of the number of people on the bus at each stop. Unfortunately, in most of the buses, the embedded sensors are only installed at front doors. Therefore, it is impossible to know exactly the number of people inside the bus. Nevertheless, the passenger counting service analyzes the bus card's IDs that each user has to validate when he goes to the bus. When the same identifier is found on multiple lines during a short time range, a connection that is followed by the corresponding passenger is detected. This information is used for updating the count of passengers inside the buses.

2.1.5. Data Recording Service

Data recorder is the service that is responsible for collecting all the information from the other services and formatting them in order to store them in a database. This service can record all internal buses's data (via the bus can), and all the data coming from the sensors, both with a given minimum frequency. A configuration file enables the user to select this frequency, and the data to back up. For performing its tasks, the data recording service is listening all the events from the other services, or from the bus can.

The data transfer between a bus and the central database is done via a mailbox mechanism proposed by the e-nove framework (`http://www.geensoft.com`). At regular intervals, the files generated by the data recorder are compacted and sent inside an email to the central server. However, the actual transfer between the buses and the central server is not achieved instantly, but only when the buses are under a WiFi coverage.

2.2. Central Server Services

Two major services are defined for managing the data storage in the central server of the considered system. They are briefly described in this section.

The data-transfer service is responsible for the daily emptying of the mailboxes that are filled by the data recording service inside each bus. This task is scheduled when all the data from the buses were provided in the mailboxes for a given working day. The data is copied in a specific directory to be treated by the data-storage service, described below.

The data-storage service performs the final step of the overall mechanism: inserting data in the central database in order to be accessible from the front-end application. It starts as soon as data from the buses is available on the central server. At this stage, the data from different files generated by the data recorder are located on the bus network, if not already done by the map matching service.

3. Bus Network Evaluation Indicators

In this section, evaluation indicators are presented. They are computed directly from the bus network model, the road map, and the data gathered in the buses.

3.1. Evaluation of the Bus Speeds

The speed of the buses is an essential information for evaluating a PTN. Indeed, the buses must not exceed the legal speed on the roads (30 or 50 km/h in France, for example). Additionally, the buses should not exceed a specific but dynamic speed over which the travel is less comfortable for the passengers.

For helping the Public Transport Authority (PTA) to notify the drivers about any problem of bus speed when they are driving, two indicators are provided:

- the average/minimal/maximal speeds of the buses on the roads traversed by each bus line; and
- the average/minimal/maximal speeds of the buses according to the time of the day for each bus line.

Let a bus line associated to a set of roads named R. The average speed of the buses for the line l is given by:

$$roadspeed(r) = \frac{\sum_{s \in E(l,r)} s}{|E(l,r)|} \tag{1}$$

where $E(l, r)$ is the set of the speed values that are measured on the road r for the buses of the line l.

Let the road segments for which an alert may be emitted to the PTA:

$$\{r | \forall r \in R, roadspeed(r) > legalspeed(r) \vee |roadspeed(r) - comfortspeed(r)| \geq \delta\} \tag{2}$$

where δ is the threshold that represents the range of the comfortable speeds. $comfortspeed(r)$ gives the preferred bus speed for obtaining a comfortable trip. This function depends on the number of passengers in the buses, and more specifically the number of standees in the bus. The definition of this function is a work under progress.

3.2. Evaluation of the Passenger Count

The evaluation of the passenger count in the buses is an essential indicator that is used by the PTA for evaluating the profitability of the bus lines, and their impacts and interests in the population activities.

Let $in(\alpha)$ the number of passengers who are boarding a bus at the bus stop α, and $out(\alpha)$ the number of passengers who are going outside the bus. These functions extract the values from the bus database. The number of passengers inside the buses for the bus line l is:

$$passengers_l(a, b) = \begin{cases} in(a) - out(a) + passenger(prev(a), a)) & \text{if } a \text{ is not the first bus stop} \\ in(a) & \text{else} \end{cases} \tag{3}$$

where $prev(a)$ replies the bus stop before the stop a in the bus itinerary.

3.3. Evaluation of Population Cover

PTA for small- and medium-size towns uses a global indicator to estimate the quality of the PTN: the population cover expressed in percent. The cover of the population is the proportion of the population of the town, which is "close" to a bus stop or a bus station. This concept of proximity is a standard national parameter of the evaluation. Basically, in France and Europe, people are assumed to be close to a bus stop when they are at most 400 meters to one of them. The population cover evaluation is based on the population density map. This map is composed of the area in which a given density of population is located. The evaluation uses the union shape of circles centered on each bus stop (see Figure 4).

Let the amount of covered populations for each circle:

$$\rho_c = \sum_{a \in \mathbb{D}, c \in \mathbb{C}} \frac{A(a \cap c).N(a)}{A(a)}, a \cap c \neq \emptyset \tag{4}$$

where $\mathbb{D}$ is the set of the polygons with the population density as an attribute; $\mathbb{C}$ is the set of the covering circles; $N(\alpha)$ is the number of people in the area α and $A(\beta)$ is the area covered by β.

For each intersecting circle pair, the intersection must be counted only one time. The covered population for the area $a \in \mathbb{C}$ is assumed to be:

$$p_a = \begin{cases} \rho_a - \dfrac{A(a \cap b).\rho_a}{A(a)} & \text{if } \exists b \in \mathbb{C} / a \cap b \neq \emptyset \\ \rho_a & \text{else} \end{cases} \tag{5}$$

Let the global population cover:

$$G = \frac{\sum_{c \in \mathbb{C}} p_c}{\sum_{a \in \mathbb{D}} N(a)} \tag{6}$$

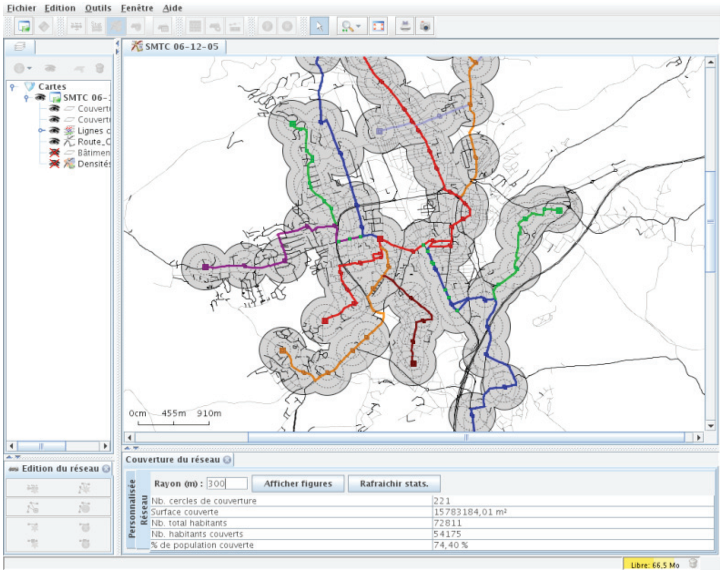

Fig. 4. Population Cover Evaluation

4. Related Works

MetroB [2] provides models, features, and tools to solve the problems related to PTN. It has user-friendly tools to edit bus networks graphically: one click to bind or unbind a road segment to a bus itinerary, add or remove bus stops, automatically create exchange stations among bus lines. MetroB computes a collection of indicators related to the structure of the bus network, e.g. average traveling time. Nevertheless, MetroB was proposed for the design and the evaluation of a bus network during its design stage. In others words, it is still difficult to evaluate a bus network during its use due to the lack of data, SARTRE aims to provide a solution to this problem by integrating a DDS system for gathering the data from buses. Additionally, SARTRE extends the collection of indicators provided by MetroB with those that are based on the data gathered in the buses.

In order to represent a busy bus operation in Kuala Lumpur city centre network, a simulation model was proposed by Moosavi et al. [3]. The calibration and validation parameters for the key bus route derive from data captured by the automatic vehicle location and the automatic fare collection system and procedures. The study is to find out what factors have highest impact on reliability of high-frequency bus services in the city centre. The simulation model is first be used to conduct a sensitivity analysis of the factors influencing reliability, such as passenger demand, terminal departure behavior, and unfilled trips.

The goal of the Digital Data Storage (DDS) specification is to facilitate the efficient distribution of data in a distributed system. Participants using DDS can "read" and "write" data efficiently and naturally with a typed interface. Underneath, the DDS middleware distributes the data so that each reading participant can access the "most-current" values. In effect, the service creates a global "data space" in which any participant can read and write. It also creates a name space to allow participants to find and share objects. DDS targets real-time systems; the API and Quality of Service are chosen to balance predictable behavior and implementation efficiency/performance. The specifications most closely related to the DDS are the OMG Notification Service [4], and the High-Level Architecture [5,6] (HLA). However, the HLA model is quite specifically designed to simulations; the data model supports a specialization hierarchy, but not an aggregation hierarchy. The set of types defined cannot evolve over time. Moreover, the data elements themselves are un-typed and un-marshalled (they are plain sequences of octets). HLA also offers no generic Quality-of-Service facilities. The OMG Notification Service suffers from technical problems for being integrated in the software architecture that was installed in the buses. Moreover, in order to not increase the cost of the communication and keep it available for the emergency and important messages, the Public Transport Authority decided to forbid real-time data connection between the buses and the central database system. These constraints imply to specify and design a dedicated architecture: the SARTRE framework.

5. Conclusion and Perspectives

In this paper, the SARTRE framework is introduced. It is an ongoing work that aims to help the PTA for designing its bus network, and supervising its operations. The framework architecture is designed for collecting data from real buses. This approach enables to have high-quality data from which indicators could be computed. These indicators are used by the PTA for planning bus service evolutions. The first version of the SARTRE framework was used for designing from scratch the bus network of the region of the city of Belfort (France, `http://www.optymo.fr`). It is currently and successfully used for collecting data from the buses.

For finalizing the SARTRE framework, several tasks should be considered. Firstly, new indicators may be proposed. For example, the PTA has expressed its need to determine and understood when, where and why the passengers are changing of a bus line, and of transport mode. These criteria enable to adapt the bus network to the usage by the population. Nevertheless, a database may be added to the framework for enabling the PTA to determine the social, economic and environmental impacts of the bus network on the city. Finally, several indicators may require some simulation-based prediction of the behaviors of the different users of the transport system (bus passengers, car drivers, bicyclists, etc.) The integration of our previous works on the urban environment modeling [7,8,9], and the individual mobility behaviors [10,11] will be considered.

Acknowledgements

The works that are presented in this paper are related to the agreement [UTBM/SAIC/SB10036PR-01] with the Public Transport Authority of the Territory of Belfort (SMTC). 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 SMTC.

References

1. Fernandez, E., De Ceach, J., Malbran, R.. Demand reponsive urban transport system design: Methdology and application. *Transportation Research Journal* 2002;**42**(7):951–972.
2. 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)*. Institute Engineering and Technology and Shanghai University; Shanghaï, China: Shanghaï University Press. ISBN 978-1-84919-326-9; 2011, .
3. Moosavi, S.M.H., Ismail, A., Golzadfar, A.. Development of simulation model to improve bus service reliability at high-frequency operation. *Jurnal Teknologi (Sciences, Engineering)* 2015;**74**(3):115–121. doi:`10.11113/jt.v74.4562`.
4. Pardo-Castellote, G.. Omg data-distribution service: architectural overview. In: *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*. 2003, p. 200–206. doi:`10.1109/ICDCSW.2003.1203555`.
5. Cai, B., Wang, C., Guan, W.S., Wang, J.. Research of information interaction simulation method in cooperative vehicle infrastructure system. In: *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. 2014, p. 45–50. doi:`10.1109/ITSC.2014.6957664`.
6. Dou, Z., Li, H.. Optimization of the border port logistics and the key-factors recognition based-on hla/sysml. *Journal of Coastal Research* 2015;:104–107doi:`10.2112/SI73-019.1`.
7. Buisson, J., Gaud, N., Galland, S., Gonçalves, M., Koukam, A.. Toward an environment for the simulation of heterogeneous entities in virtual cities. In: *International Workshop on Environments for Multiagent Systems (E4MAS14)*. IFAAMAS; Paris, France: Springer; 2014, .
8. Galland, S., Balbo, F., Gaud, N., Rodriguez, S., Picard, G., Boissier, O.. Contextualize agent interactions by combining social and physical dimensions in the environment. In: Demazeau, Y., Decker, K., editors. *13th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS), LNAI 9086*. Springer International Publishing Switzerland; 2015, p. 107–119. doi:`10.1007/978-3-319-18944-4{\string_}9`.
9. Rodriguez, S., Galland, S., Gaud, N.. A new perspective on multi-agent environment with sarl. In: Bordini, R., Elkind, E., editors. *14th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS15)*. Belfort, France; 2015, .
10. Galland, S., Buisson, J., Gaud, N., Gonçalves, M., Koukam, A., Guiot, F., et al. Agent-based simulation of drivers with the janus platform. In: *3nd International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTRANS14)*. Hasselt, Belgium: Elsevier; 2014, p. 738–743. doi:`10.1016/j.procs.2014.05.484`; procedia Computer Science, vol. 32.

11. Galland, S., Knapen, L., Yasar, A.u.h., Gaud, N., Janssens, D., Lamotte, O., et al. Multi-agent simulation of individual mobility behavior in carpooling. *International Journal on Transport Research Part C* 2014;**45**:83–98. URL: http://www.sciencedirect.com/science/article/pii/S0968090X14000035. doi:10.1016/j.trc.2013.12.012.