Simulating the Impact of Future Internet on Multimodal Mobility

Mahdi Zargayouna[1], Besma Zeddini[2], Gérard Scemama[1], Amine Othman[1]

[1] Université Paris-Est, IFSTTAR, GRETTIA
Champs sur Marne, France.

[2] LARIS, EISTI
Cergy-Pontoise, France.

{hamza-mahdi.zargayouna, gerard.scemama, amine.othman}@ifsttar.fr, bzi@eisti.fr

***Abstract*—In the context of the EC-funded project Instant Mobility, we have defined a comprehensive architecture for transport and mobility applications that aim to innovate by introducing future Internet technologies to this domain. We have defined an Internet-based "multimodal travel platform" that provides information and services able to support new types of connected transport applications. The considered scenario is centered on multimodal travelers (both drivers and passengers). In this paper, we present the SM4T[1] simulator, that we have designed and implemented to test the platform and to interact with it on behalf of multimodal travelers. SM4T is a fully agent-based simulator for multimodal travelers mobility. The application simulates the movements of travelers on the different transport networks (road and public transport) while taking into account the changes in travel times and the status of the networks.**

Keywords*—*Social Computing, Multiagent Simulation, Future Internet, Transport, Applications

I. Introduction

The Internet of networks, web services, cloud computing, social networks and Internet of Things provide new possibilities for transport applications, which were not imaginable a few years ago. Using future Internet technologies, we are more and more able to provide real-time information and up-to-date mobility solutions for multimodal journeys. Travelers are able to access up-to-date information on transport modes, their availability, ticketing, operation situation, etc., allowing the planning of an optimized and up-to-date itinerary following a great number of criteria. It will soon be possible to connect all transport modes and vehicles whenever needed to assist a traveler during a multimodal journey, informing him about any disruption and delivering a choice of alternative solutions.

In the context of the EC-funded project Instant Mobility, we have designed and implemented the multimodal travel platform (MMT) [15] that guides the traveler to the needed transit stops/stations or directs the car driver along the best route. We will briefly describe the platform, which supports the provision of the essential subset of these services. We mostly present the multimodal travel simulator SM4T (Simulator for Multiagent MultiModal Mobility of Travelers) that interacts with the platform on behalf of travelers and transport means and allows for the understanding of future status of the networks. SM4T enables for the rapid prototyping and execution of simulations for several kinds of online applications. The application simulates the movements of travelers on the different transport modes and networks while taking into account the changes in travel times and the status of the networks. Since it assumes the continuous localization of travelers, SM4T can notably simulate and evaluate the impact of a wide range of community transportation apps, such as user-submitted travel times and route details, community-based driver assistance, community parking, etc.. This paper present the main building blocks of the platform and the simulator, which we believe provides design principles making them reproducible in other transport applications pursuing similar objectives.

The multiagent paradigm is relevant for the simulation of urban transport systems. It indeed facilitates an approach by analogy in the transport domain which one of the objectives is the coordination of distributed entities. This is why the multiagent approach is often chosen to model, solve and simulate transport problems [3]. This approach is particularly relevant for the simulation of travelers mobility since the objective is to take into account human behaviors that interact in a complex, dynamic and open environment. The proposed simulator is composed of autonomous agents, evolving in an environment on which they have a partial perception.

The remainder of this paper is structured as follows. In section II, we discuss the choice of the simulation platform and previous proposals for travelers mobility simulation. Section III presents the MMT platform. In section IV, we detail the simulator design: its parameters and data, the behaviors of the agents in the system, the multimodal path planning and the representation of the multimodal networks. In section V, we describe our experiments before to conclude and describe some further work we are conducting.

II. Related Work

To design and implement a multiagent simulator, it is possible to develop an application directly in a host

[1] Simulator for Multiagent MultiModal Mobility of Travelers

programming language. However, it is often faster, more useful and more efficient to ground the simulator on an existent multiagent simulation platform. In the context of transportation applications, one main choice criterion for the simulation platform is its ability to create geospatial agent-based models, i.e. its ability to integrate and process geographic data. Based on this criterion, we haven't considered several popular multiagent platforms such as Jade [1], Mason [7] and Madkit [4], for which it is difficult to integrate GIS capabilities.

Swarm [9] is a simulation platform that has a library allowing for the loading of layers of GIS data. However, it does not provide spatial primitives nor gives the possibility to store the resulted environment [12]. Netlogo allows to import and export GIS data and provides some basic geometrical operations, but not the more advanced spatial analysis operations. Gama [12] provides an environment for building spatially explicit agent-based simulations, while Repast Simphony [13] integrates a GIS library (Geotools), and provides additional GIS services (network modeling as a graph, computation of shortest paths, visualization and management of 2D and 3D data, etc.). We have chosen the Repast Simphony platform, since between all the available simulation platforms that fit with our requirements, it is the most mature one and more importantly, it integrates several Api to deploy simulators over several hosts[2].

Besides, there exists several multiagent simulators for travelers mobility. For instance, MATSim [8] is a widely known platform for mobility micro-simulation. However, the mobile entities in MATSim are passive and their state is modified by central modules, which limits its flexibility and its ability to integrate new types of (proactive) agents. Transims [10] simulates multimodal movements and evaluates impacts of policy changes in traffic or demographic characteristics while Miro [2] reproduces the urban dynamics of a French city and proposes a prototype of multiagent simulation that is able to test planning scenarios and to specify individuals' behaviors. AgentPolis [6] is also a multiagent platform for multimodal transportation. However, none of these proposals assume the continuous localization of travelers and means of transport. In addition, none of them integrate dynamic ridesharing as one of the transport modes in the multimodal network. In the proposal described in this paper, travels are fully multimodal: they concern private cars, all the public transport modes as well as ridesharing and pedestrians. Carpooling, car and bicycle sharing services are easily integrable in the simulator. Travelers routes are continuously monitored and alternatives are proposed to them if something wrong happens with their itinerary.

III. The Multimodal Travel Platform

In this section, we provide a general description of the MMT platform, which is necessary to understand the considered scenario and the development of the simulator. The enablers complete specifications and the technical details of the implemented platforms can be found in the Instant Mobility project public deliverables(e.g. [14]).

A. Scenario and assumptions

In the scenario that we consider, online services offer a traveler a wide range of personalized travel and transport options, according to his preferences. The trip may use various modes including public transport, car, and ride sharing. When planning a given journey, the traveler receives a number of alternative integrated itineraries which are the optimal plans that respect his preferences and take account of the latest real-time information from all relevant modes such as transit routes, timetables, vehicles positions, fare costs, available ride sharing opportunities, etc. Once it receives confirmation of the chosen itinerary, the travel assistant application books the corresponding services, and delivers an integrated ticket receipt while requesting a single payment from the traveler's account. The traveler receives his chosen itinerary via Internet, and throughout the journey is helped to get off at the right stops, walk through a complex terminal or interchange, find the next transport link, etc. During the journey, the itinerary is continuously monitored in real-time, and the traveler is alerted whenever conditions or options change. The current itinerary is then updated according to his choice. The traveler is charged at the end of his trip. That means that if there is a problem with the planned journey, it is taken into account, and the user won't have to ask for a refund in case something changes. He also only pays for the service he has asked for. That means that a traveler that uses only one means of transport would only pay for that service.

We make two main assumptions in the context of the MMT platform. On one hand, we assume that every traveler's mobile Internet device is aware of its location and every transport vehicle (public and private) is tracked continuously, outdoors and indoors. On the other hand, the platform is able to work with both pre-trip reservations as well as to fulfil its primary objective, to manage changes to the itinerary in real time, incorporating any new traveler requests as well as traffic incidents and transport service deviations. The scenario is formalized with the use case diagram of Figure 1.

The main challenge with this scenario and these assumptions is twofold:

- From a technological perspective, to find the right architectures and system functionalities to take full advantage of the future Internet technologies, while respecting travelers' preferences and data privacy needs;
- From an algorithmic perspective, to find the right procedures and balancing mechanisms to keep each traveler on what is for him the optimum and preferred itinerary while avoiding adding to traffic congestion and overcrowding on public transport.

B. Public interface

Figure 2 presents a view of the multimodal travel platform from an external point of view. It specifies the requirements for a specific region to set an Instant Mobility service and the different exchanges that should take

[2] The distribution of our simulator over several hosts is one of our ongoing works.

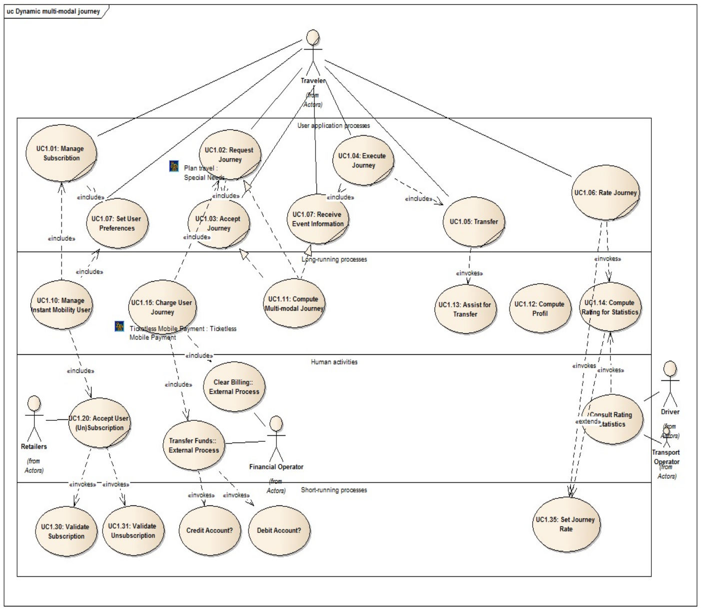
uc Dynamic multi-modal journey
Traveler
(from Actors)
User application processes
UC1.01: Manage Subscribtion
UC1.02: Request Journey
Plan travel : Special Needs
UC1.04: Execute Journey
UC1.06: Rate Journey
«include»
UC1.07: Set User Preferences
UC1.03: Accept Journey
UC1.07: Receive Event Information
UC1.05: Transfer
Long-running processes
«invokes»
UC1.10: Manage Instant Mobility User
UC1.15: Charge User Journey
Ticketless Mobile Payment : Ticketless Mobile Payment
UC1.11: Compute Multi-modal Journey
UC1.13: Assist for Transfer
UC1.12: Compute Profil
UC1.14: Compute Rating for Statistics
Human activities
Clear Billing:: External Process
Retailers
(from Actors)
UC1.20: Accept User (Un)Subscription
Transfer Funds:: External Process
Financial Operator
(from Actors)
Consult Rating Statistics
«extend»
Driver
(from Actors)
Transport Operator
(from Actors)
Short-running processes
UC1.30: Validate Subscription
UC1.31: Validate Unsubscription
Credit Account?
Debit Account?
UC1.35: Set Journey Rate

Fig. 1. Scenario [14]

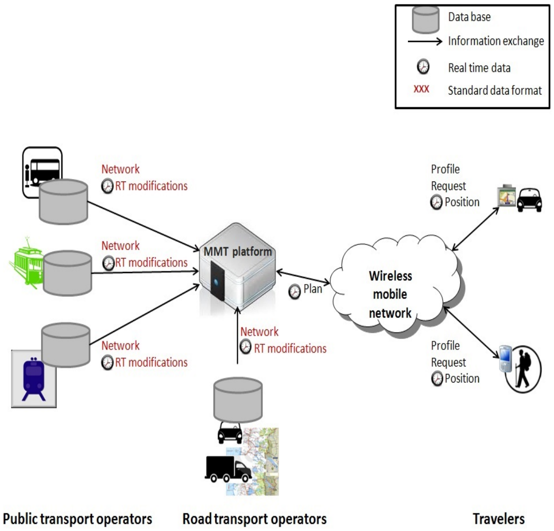

Fig. 2. Multimodal travel platform public interface

place with the platform. The platform interacts with three types of actors: the public transport operators, the road transport operators and the travelers.

Each public transport operator has to provide the platform a description of their network and theoretical timetables. As depicted in the figure, we do not require the public transport operators to have a common database that integrate all the public transport means and networks. Each transport mode operator might have its own database, and the platform has to integrate them and manage them simultaneously. Each public transport operator has to provide three types of data:

- The description of the transport offer
- The position, the advance/delay of all its fleet

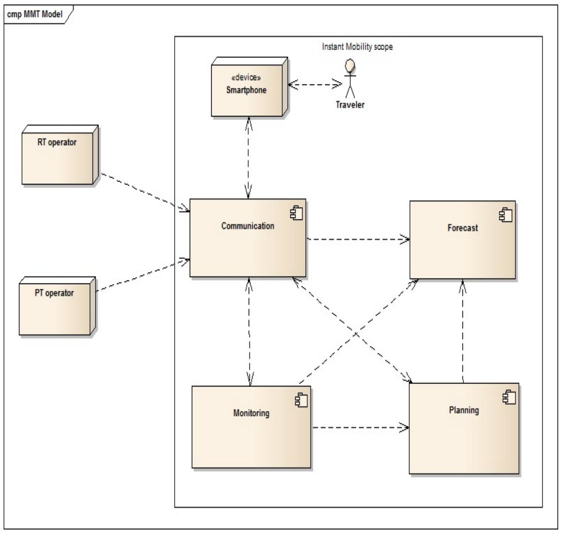

Fig. 3. The MMT platform model

- The events that cause transport services disruptions

As for public transport operators, road transport operators have to provide the MMT platform with a description of their network, together with all static information related to it. They also have to provide three types of data:

- The description of the road network
- The speeds, densities, occupancy rates or status
- The events which impacts the transport offer

All these exchanged data have to conform to the latest European standards.

The travelers interact with the platform using standard communication protocols. Each traveler provides the system with his profile, which includes detailed information regarding his properties and preferences. Once a plan is received from the platform, the mobile device of the user dynamically sends his current position to the platform. If the difference between the actual position of the traveler and the planned position (i.e. the expected position following the plan that was proposed to the traveler) is big enough, following the traveler's preferences, the traveler receives a new plan taking into account his new context.

C. Model

We define a high-level representation of MMT from a functional point of view. It comprises four modules designed as independent and loosely coupled components (cf. Figure 5):

- The communication module is responsible for the interaction of MMT with the outside world (road transport operators, public transport operators, and travelers, cf. Figure 4);
- The planning module is responsible for planning and re-planning of travelers' itineraries (see Figure 5);

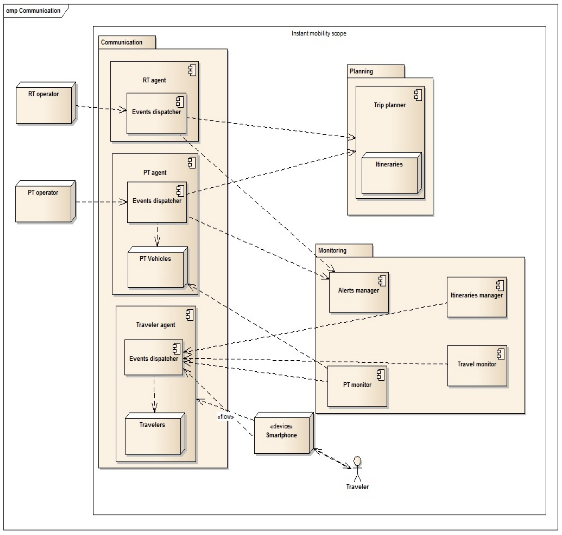

Fig. 4. The communication model

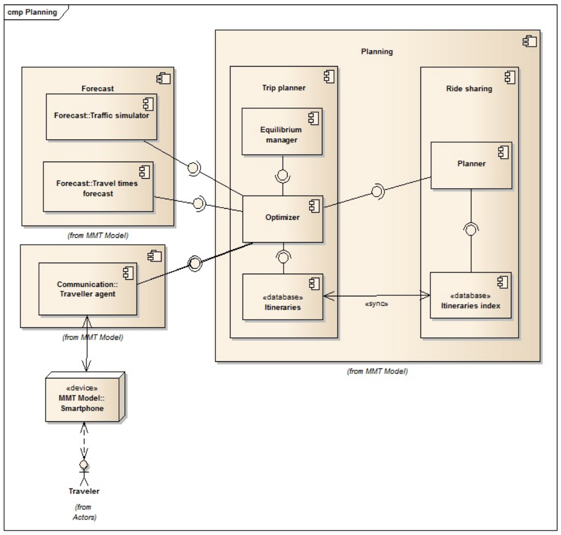

Fig. 5. The planning model

- The forecast module maintains the best possible vision of the future status of the networks. The planning module bases its calculation on the outcome of this module;
- Finally, the monitoring module monitors the execution of the travelers' plans.

The mobile device of the traveler could store information that would speed-up the calculation and/or improve the quality of the solutions provided to him. For instance, it could store information about users' travel habits. This would save a lot of users' time by immediately making suggestions and recommendations to him.

In order to function properly, the MMT platform has to interact with the outside world (cf. the public interface described earlier) via its communication module described earlier as well. Since it is very difficult to test this scenario under real conditions, with notably a significant number of multimodal travelers equipped with smartphones and an Instant Mobility app, we design and implement the SM4T simulator. Its purpose is to completely replace the environment of the platform and provide it with all the needed data: static and dynamic, concerning the transport operators and the travelers all along their journey.

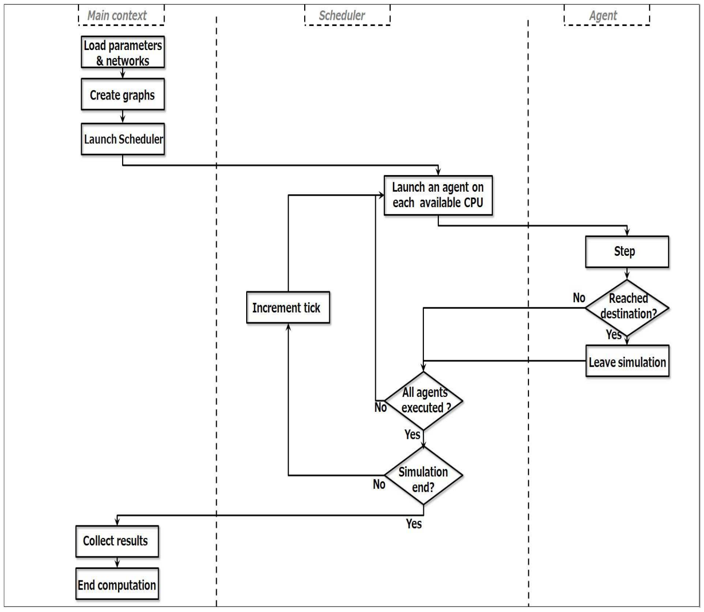

Fig. 6. Workflow of the simulation

IV. The Simulator

The SM4T simulator is implemented on top of the Repast Simphony multiagent platform [13]. Its purpose is to represent travelers (drivers and passengers) and transport means (public transport vehicles and private cars) in a micro-level and to simulate their dynamic movements and their interaction with the MMT platform (tracking, planning requests, plans update, etc.). To this end, it needs input data that are similar to those needed by the platform, and some additional parameters.

The simulator can be used in isolation from the platform, in which case the exchanges with the platform are turned off. To this end, some functionalities of the platform are cloned and integrated to the simulator.

A. Data and Parameters

The workflow of Figure 6 details one simulation execution and structures the remainder of the presentation. The simulation starts with the loading of the parameters (simulation duration, number of agents of each type, the default speeds of each agent type, etc.). Then, the main context program creates the logical graphs (described in section IV-C) and launches the scheduler. The scheduler is responsible of the synchronization of execution of the agents per tick of simulated time. Repast comes with a discrete event scheduler. The default scheduler iterates over the agents and executes their *step* method, describing the action they intend to execute during a tick of simulated time. We have replaced this scheduler by a new one that instead launches the active agents in parallel over all the available CPUs. When launched, each agent executes a step method. The behavior of each agent type is described in section IV-B. Finally, when the simulation duration is reached, the results are reported and the simulation ends. All along the simulation, the agents interact with the MMT platform described in the previous section and provide it with their profiles, preferences, service requests and real-time positions.

1) Data: The input data (xml files) of the simulator are:

- the road network,
- the public transport network,
- the transfer mapping,
- the timetables of the public transport vehicles,
- the pedestrian network,
- the travel patterns,
- the travelers profiles.

The road network is a description of the roads, crossroads and driving directions. Each road has, among other information, a corresponding minimum and maximum speeds. It also has a mapping between traffic flows (vehicles/hour), the traffic density and speeds. A public transport network is composed of transport lines, each of them composed of a set of itineraries. An itinerary is composed of a sequence of edges. Each edge has a tracing in the form of a sequence of pairs $\langle longitude, latitude \rangle$, and is composed of an origin node and a destination node. Finally, every node is defined by its name and coordinates. The transfer mapping is a table informing about the stops of the network for which a transfer by foot is possible and the road transport nodes that are reachable from the stops. The timetables of the vehicles are composed of a set of missions. Each mission corresponds to a specific itinerary[3] and describes the path of a vehicle and the corresponding visit times. Each timetable is then a sequence of pairs $\langle stop, time \rangle$. The pedestrian network is a subset of the road network in which pedestrians can move, but which is undirected (pedestrians don't have to obey to the one-way limitations). A travel pattern clusters the considered geographic region in zones and describes the number of persons asking to leave or to join each region. The travelers profiles define the properties and preferences of the travelers. Most importantly for the simulator, they define which drivers and which passenger are interested by the ridesharing service. The preferences of the traveler also define the accepted time gap between their computed itinerary and their real situation, before asking for a new up-to-date itinerary.

2) Parameters:

a) Simulation duration: Two values define the duration of a simulation run. The first is an interval defining the first and last date that are simulated (noted τ^- and τ^+ respectively). In the absence of these parameters, we use respectively the values associated to the maximum date and minimum date in the timetables of the public transport vehicles. The second value that has to be defined is the number of discrete ticks of time that the simulation will execute before terminating (noted δ henceforth). At each tick, all the agents are activated for a particular action defined by their behavior.

b) Agents speeds: The speeds of the agents are defined in three ways:

- Public transport vehicles: based on the input data, they are inferred from the $\langle stop, time \rangle$ pairs.
- Private cars: based on the maps data, their speeds are taken from the mapping density/speed of the road they are currently traversing.
- Default: the simulator user defines speeds for pedestrian, cars and public transport vehicles as a parameter. The user-defined mean car speed is used if the speed data is missing from the current road of the car. The user-defined public transport speed is used if two successor pairs $\langle stop, time \rangle$ give an inconsistent speed due to errors in the data[4].

c) Units transformation: All the data that are expressed in function of time (e.g. the visit times of the public transport vehicles) have to be expressed in terms of simulation ticks. However, since the original time data are expressed in terms of date, they have to be transformed as follows. Let t the original time, t' is the new time (in simulation tick), computed this way:

$$t' = \frac{t - \tau^-}{\gamma} \times \delta, \text{with } \gamma = \tau^+ - \tau^-$$

In addition, all the speeds are originally defined in terms of Km/h. They have to be transformed into meters/tick as follows (σ_{mode} is the speed of the transport mode, and $\sigma_{modeKmH}$ is its original speed expressed in Km/h).

$$\sigma_{mode} = \frac{\sigma_{modeKmH} \times \gamma}{3.6 \times \delta}$$

Let us take the private car mode as an example. $\frac{\gamma}{\delta}$ gives the number of seconds elapsed in a simulation tick of time. Thus, $\sigma_{car} = \frac{\sigma_{carKmH} \times \gamma}{3.6 \times \delta}$ gives the number of meters that the car can travel in a tick of simulation time. The same principle applies for public transport vehicles and pedestrians.

[3] This correspondence mission-itinerary has to be kept in mind as we will be referring to it in section IV-C.

[4] E.g. infinite speed due to two identical visit times of two successive stops.

B. Multiagent System

1) Planner Agents: The planner agents have the responsibility of computing the best road itinerary for the car agents and the best multimodal itinerary for the traveler agents. A planner agent is created when an agent request is submitted to the system, and leaves the system right after. Planner agents interact with the platform on behalf of the travelers and private cars.

The planning module of the MMT platform bases its computation on the latest status of the networks. The graph representation and the paths calculations are described in section IV-C. A road plan is composed of a sequence of edges together with their corresponding visit times. A multimodal itinerary is composed of a sequence of pairs $\langle id_{vehicle}, itinerary\rangle$, with $id_{vehicle}$ the identifier of the vehicle to take and the corresponding part of the itinerary. This capability is cloned in the simulator, in case SM4T needs to be used independently from the platform.

2) Agents Movements: Each active agent has a list of coordinates that he has to follow, resulting from the itinerary that he received from a planner agent. Agents are allowed to move for a certain distance at each tick equal to σ_{car}, $\sigma_{PTvehicle}$ or $\sigma_{passenger}$ depending on the type of agent. At each tick, the agent iteratively checks if he can move from his current coordinate to the next in his list. If not, he calculates the intermediate coordinate that corresponds with the remaining distance that he is allowed to travel. In the next tick, the agent can travel the remaining distance to the next coordinate. When the MMT platform notifications are on, the agents send their positions to the platform at each time tick.

3) Car Agents: When created, a car agent has an origin and a destination. If travel patterns exist for the considered network (cf. section IV-A), the origin and destination of each car agent are chosen such as the origins and destinations of all agents are proportional to the pattern. If not, the origin and destination of the car agent are chosen randomly. The agent then asks the planner for the best itinerary between his origin and his destination. If the driver profile states that the corresponding car agent is interested in ridesharing, the car agent registers himself in the nodes of his itinerary (cf. section IV-C for the use of these registrations). If there are passengers onboard, sharing their ride with the driver, they are moved to the same coordinates at the same time by the car agent. That means that, when they are onboard a a car, traveler agents delegate the control of their movements to the car agent. The same principle applies for the public transport vehicles. At each simulation tick, the car agent checks if he has reached his destination. If so, he leaves the simulation.

4) Public Transport Vehicle Agents: The public transport vehicle agents don't choose their origin and destination and obey to predefined timetables. Each vehicle agent when created, infers his itinerary from his timetable. While the vehicle agent has not reached his destination, he travels at each tick the allowed distance, following his current speed. When the vehicle reaches a stop, he looks in his onboard travelers who has to leave at this stop. Then he looks in the waiting travelers at the stop who has to take him.

5) Traveler Agents: As for car agents, the origin and destination of the traveler agent are either inferred from travel patterns if they exist, or are chosen randomly. When they are not walking, traveler agents do not travel on their own, but share rides with other drivers and/or take public transport vehicles, which are responsible of their movements. The traveler agent alternates between walking and waiting for a vehicle (car or public transport).

6) Itinerary Monitoring: Each traveler agent and each car agent have their their itinerary monitored. The monitoring purpose is to ask for a new itinerary in two cases:

1) the real position of the agent is different from the planned one with a certain gap Δ_a (defined in the preferences of the agent);
2) there is an event on the transport network that impacts the traveler or the car planned itinerary.

The first case is handled by a behavior of the agents, since both the position of the agent and his plan are part of his knowledge.

The second case however needs an interaction with the MMT platform. Indeed, the platform monitors the itineraries of the travelers (cf. Figure 5). To be aware of the only events that concern the agent, the monitoring module of MMT subscribes the traveler to the only edges of the transport network that form his itinerary. When the travel time of an edge changes, the new travel time is broadcasted to the subscribing travelers. The planning process is then launched with the new travel times associated with the network. This feature is cloned and deported to the simulator for the standalone use.

C. Multimodal Itinerary Planning

The planning module of MMT is responsible for computing the itineraries for cars and passengers. To this end, it executes the A-Star algorithm [5] on the road and multimodal networks. The value of h - the heuristic - associated to each vertex v for a specific destination v_d is equal to the geodesic distance between v and v_d divided by the maximum speed of the vehicles in the system. Recall that for a heuristic to be valid in A-Star, it doesn't have to be overestimated, which is verified for the values that we have chosen. However, for passengers, the computed itinerary could be the shortest in terms of theoretical travel time, but would result in too many transfers if the computed path contains too many different itineraries. Solving this problem is the purpose of the next subsection.

1) Public Transport Itinerary Planning: In order to avoid having passengers' shortest paths that pass by too many itineraries and thus result in too many transfers, we modify the public transport graph as follows (cf. Figure 7). Let $\langle s_1, s_2\rangle$ an edge in the public transport graph. Let it_1 and it_2 two itineraries passing by this edge. We create four new vertices s'_1, s'_2 that we connect with the edge that belongs to it_1 and s_1" and s_2" that we connect with the edge that belongs to it_2. We also create four new edges $\langle s'_1, s_1"\rangle$, $\langle s_1", s'_1\rangle$, $\langle s'_2, s_2"\rangle$ and $\langle s_2", s'_2\rangle$, that we note pedestrian edges and that are heavily penalized. When we run a

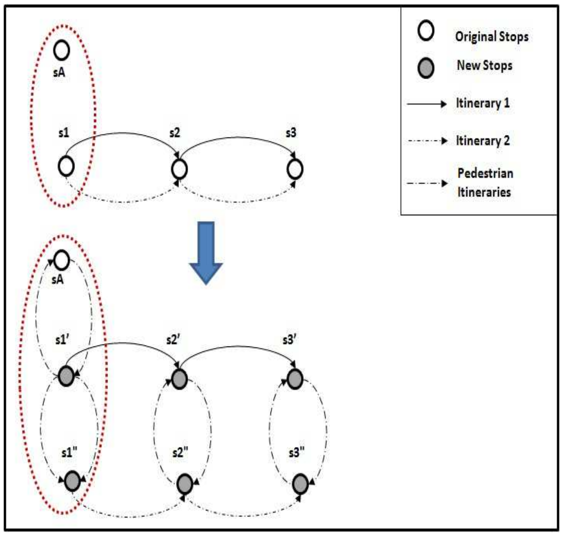

Fig. 7. Multimodal Network Transformation

shortest path algorithm on this new graph, the itineraries that are on the same vehicle/itinerary are encouraged and a transfer is only proposed when it is either impossible or really expensive to stay on the same vehicle.

The transfer map is modified so that the stops or the crossroads that are reachable from s_1 (s_A in the figure) by foot become reachable from s'_1 and s_1” (the read dotted circle in the figure). After applying the A-Star shortest path algorithm on the resulting itinerary, the planning module interrogates the stops of the best found itinerary to infer the sequence of vehicles that the traveler agent will have to take and sends back the result to the traveler agent.

2) Ridesharing: We consider a ridesharing service that is designed as a complementary service to public transport. We believe that this increases the chances for ridesharing to be a successful service. We consider that a passenger would hardly accept to take more than one private car in a single itinerary. Based on these assumptions, if the passenger preferences indicate that he would be interested in ridesharing, the planning module of MMT will look for these types of itineraries, following this order of priority:

1) Find a ridesharing itinerary with a single vehicle from the origin to the destination
2) Find a ridesharing itinerary from the origin followed by a public transport itinerary, or a public transport itinerary followed by a ridesharing itinerary to the destination
3) Find a public transport itinerary

Let C_o and C_d the cars that are registered in node o and node d (resp. the closest node to the origin of the traveler and the closest node to the destination of the traveler). To see if there is a ridesharing itinerary with a single vehicle from the origin to the destination, we calculate $C = C_o \cap C_d$. If $C \neq \emptyset$, the car $c \in C$ that respects the time constraints of the traveler and that arrive the soonest to d is chosen and his itinerary sent to the traveler agent. If no such vehicle is found, i.e. no car can transport the traveler from his origin to his destination directly, we look for itineraries which start or finish with a public transport itinerary. To this end, let $it_t = v_1, \ldots, v_n$ the itinerary of the traveler agent. The planning module computes iteratively the intersection of the vehicles registered in the successive nodes starting from v_1 looking for the vehicle that takes the traveler the furthest while respecting his time constraints. The same process is repeated starting from v_n and backwards looking for the vehicle that can transport the traveler to his destination. If such vehicle is found, the remainder of the itinerary is computed on the public transport network as explained in the previous paragraph. Finally, if none of these calculations succeeds, the shortest path in public transport is calculated. Again, this capacity is also present in the simulator for a standalone use.

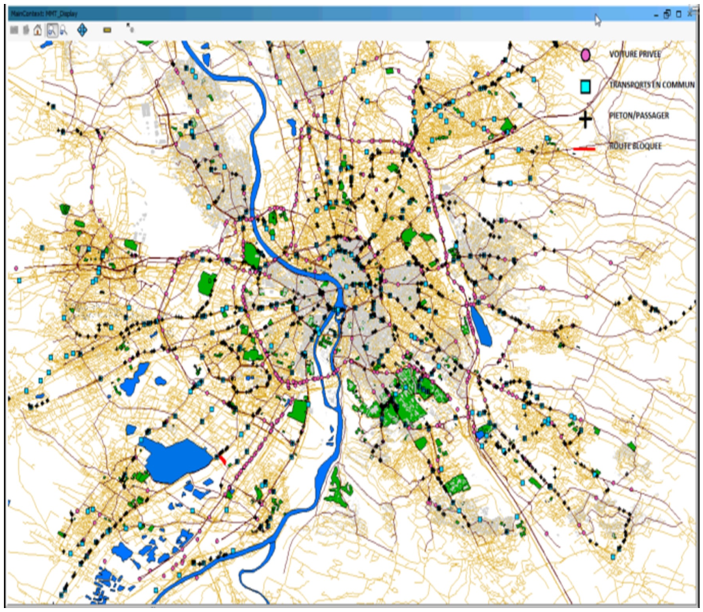

Fig. 8. Simulation Execution

V. Experiments and results

We demonstrate the use of the platform for the city of Toulouse, France (cf. Figure 8), for which we have detailed data, including urban travel patterns [16]. We have considered the main roads of the transport network of Toulouse with about 18,000 roads. We also have considered the public transport network of Toulouse, with 80 lines, 359 itineraries and 3,887 arcs. In our current simulations, our multiagent system is made of 118,270 agents: 28,720 buses[5], 30,000 cars, 30,000 drivers and 30,000 passengers. Travel times are updated with real data from the ClaireSiti platform [11].

Since the transport network parameters feed our simulator, the positions of our agents are compliant with the real life situation. Thanks to the simulator, the platform has been used for a demonstration of the Instant Mobility project in the ITS World Congress 2012. Together, they

[5] In our simulations, we create one bus per mission.

have provided a visual tool allowing for the follow-up of ongoing individual trips. The demonstration assesses the impact of the platform use on the multimodal traffic quality, notably under unusual traffic conditions (breakdowns, accident, etc.). The demonstration also validates the effect of the platform on the actual use of ride-sharing service. By making it easier for travelers and car drivers to meet and share rides, the demonstration predicts that users would have more incentives to actually use this service. Itineraries including ridesharing are well integrated with the multimodal itineraries and up to 20% of the travelers actually use the ridesharing service. Furthermore, with the capacity of the simulator to inform travelers about disturbances and to provide alternatives, the preliminary results shows an improvement of 15% in travel time for informed travelers as compared to uninformed ones.

VI. Conclusion and Perspectives

In this paper, we have described the main building blocks of a multimodal platform and an agent-based simulator for multimodal travelers, which are easily reproducible in other transport applications pursuing similar objectives. We have presented the main functionalities together with the data and parameters necessary for both applications to work properly, with a focus on the simulator.

SM4T now simulates individual spatiotemporal positions that are compatible with the real situation. It is then possible to integrate transport applications that use these positions to provide an advanced service. Indeed, even if the primary objective of SM4T was the testing of the MMT platform, it uses go far beyond it. Indeed, SM4T now enables for the rapid prototyping and execution of simulations for several kinds of new online applications that need to track individual travelers such as route guidance, urban parking assistance, user-submitted travel times and route details, community-based driver assistance, etc.

In the near future, we plan to develop scenarios with hundreds of thousands of passengers and drivers to verify that our implementation is actually scalable. The distribution of the simulator on the cloud is also currently conducted. Our second ongoing research is a consequence of the success of the first one: when we simulate realistic number of agents in the transport network (say, millions of travelers), the problem of equilibrium arises. Indeed, if we send too many cars or travelers to the same itineraries, we risk to create congestions as a direct consequence of our guidance service. To avoid this drawback, we have to assign travelers and cars to itineraries while respecting their distribution over competitive paths and not send them all to the same roads or vehicles.

Acknowledgement

This paper reflects only the authors' views; the European Union is not liable for any use that may be made of the information contained therein. The work reported is carried out by the Instant Mobility project, which has received research funding from the European Union, under the FP7-2011-ICT-FI programme.

References

[1] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE.* Wiley, 2007.

[2] S. Chipeaux, F. Bouquet, C. Lang, and N. Marilleau. Modelling of complex systems with AML as realized in MIRO project. In *LAFLang 2011 workshop*, pages 159–162, Lyon, France, 2011. IEEE Computer Society.

[3] P. Davidsson, L. Henesey, L. Ramstedt, J. Tornquist, and F. Wernstedt. An analysis of agent-based approaches to transport logistics. *Transportation Research Part C - Emerging Technologies*, 13(4):255–271, 2005.

[4] O. Gutknecht and J. Ferber. The madkit agent platform architecture. In *Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55, London, UK, UK, 2001. Springer-Verlag.

[5] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2):100–107, 1968.

[6] M. Jakob, Z. Moler, A. Komenda, Z. Yin, A. X. Jiang, M. P. Johnson, M. Pechoucek, and M. Tambe. Agentpolis: towards a platform for fully agent-based modeling of multi-modal transportation (demonstration). In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pages 1501–1502, 2012.

[7] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, July 2005.

[8] M. Maciejewski and K. Nagel. Towards multi-agent simulation of the dynamic vehicle routing problem in matsim. In *Proceedings of the 9th International Conference on Parallel Processing and Applied Mathematics - Volume Part II*, PPAM'11, pages 551–560, Berlin, Heidelberg, 2012. Springer-Verlag.

[9] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The swarm simulation system: A toolkit for building multi-agent simulations. Technical report, Santa Fe Institute, 1996.

[10] K. Nagel and M. Rickert. Parallel implementation of the transims micro-simulation. *Parallel Computing*, 27(12):1611–1639, 2001.

[11] G. Scemama and O. Carles. Claire-siti, public road transport network management control: a unified approach. In *Proc. of 12th IEE Int. Conf. on Road Transport Information & Control (RTIC'04)*, pages 11–18, London (UK), April 2004.

[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 *PRIMA*, volume 7057 of *Lecture Notes in Computer Science*, pages 242–258. Springer, 2012.

[13] E. Tatara and J. Ozik. How to build an agent-based model iii – repast simphony. In *Applied Agent-based Modeling in Management Research, Academy of Management Annual Meeting, Chicago*, 2009.

[14] M. Zargayouna. Multimodal journey optimisation enablers specifications. Technical report, Instant Mobility Project, 2012.

[15] M. Zargayouna, G. Scemama, B. Zeddini, P. Kompfner, P. Gatellier, P. Constant, and D. Beckman. Future internet for a personal travel companion service. In *19th ITS World Congress*, 2012.

[16] M. Zargayouna, B. Zeddini, G. Scemama, and A. Othman. Agent-based simulator for travellers multimodal mobility. *Frontiers of Artificial Intelligence and Applications*, 252:81–90, 2013.