Agent-Based Simulator for Travelers Multimodal Mobility

Mahdi ZARGAYOUNA [a,1], Besma ZEDDINI [a], Gérard SCEMAMA [a] and Amine OTHMAN [a]

[a] *Université Paris-Est, IFSTTAR, GRETTIA, France*

Abstract. Transport systems are more and more complex and have to evolve to integrate more connected entities (mobile devices, localized vehicles, etc.). It becomes critical to develop micro-simulation tools for mobility policies makers taking into account this fact. In this paper, we propose a multimodal travel simulator that allows for the understanding and the prediction of future status of the networks and allows for testing new online applications. The application simulates the movements of travelers on the different networks while taking into account the changes in travel times and the status of the networks. The considered transport modes include pedestrians, private cars, all public transport modes and ridesharing. The simulator has been developed using the Repast Simphony® multiagent simulation platform.

Keywords. Transportation, Multimodal services, Modeling, Simulation.

1. Introduction

Multimodal traffic information and control platforms such as [1] integrate different information sources about different transport modes. They allow for a common visualization of the different networks states, and the followup of several indicators (regularity, advance, delay, etc.) related to the different networks components (public transport lines, itineraries, roads, etc.). However, current platforms provides rarely integrated simulation tools that would predict the effect of control actions on the multimodal networks. More generally, transport systems are becoming more and more complex and have to evolve to integrate more connected entities (mobile devices, localized vehicles, trackable goods, etc.), and it becomes critical to develop micro-simulation tools for mobility policies makers capable of representing and managing these entities.

In this paper, we propose a multimodal travel simulator that allows for the understanding and the prediction of future status of the networks and that can be used for testing new online applications that track individual travelers. Online applications are applications that track individual travelers and need to be con-

[1]Corresponding Author: Mahdi Zargayouna, Université Paris-Est, IFSTTAR, GRETTIA, Boulevard Newton, Champs sur Marne F-77447 Marne la Vallée Cedex 2, France; E-mail: hamza-mahdi.zargayouna@ifsttar.fr.

tinuously aware of their positions. Current tested applications are urban parking, dial-a-ride and crisis management. The application simulates the movements of travelers on the different networks while taking into account the changes in travel times and the status of the networks. In this paper, we present the main building blocks of the simulator, which are easily reproductible in other simulators 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 [2]. 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 of which they have a partial perception.

The remainder of this paper is structured as follows. In section 2, we discuss the choice of the simulation platform and previous proposals for travelers mobility simulation. Section 3 presents the parameters and data of the simulator. Section 4 describes the behaviors of the agents in our system. In the section 5, we detail the multimodal path planing and the representation of the multimodal networks before to conclude and describe some further work we are conducting.

2. Related Work

To design and implement a multiagent simulator, it is possible to develop an application directly in a host 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 [3], Mason [4] and Madkit [5], for which it is difficult to integrate GIS capabilities.

Swarm [6] 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 [7]. Netlogo allows to import and export GIS data and provides some basic geometrical operations, but not the more advanced spatial analysis operations. Gama [7] provides an environment for building spatially explicit agent-based simulations, while Repast Simphony [8] 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].

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

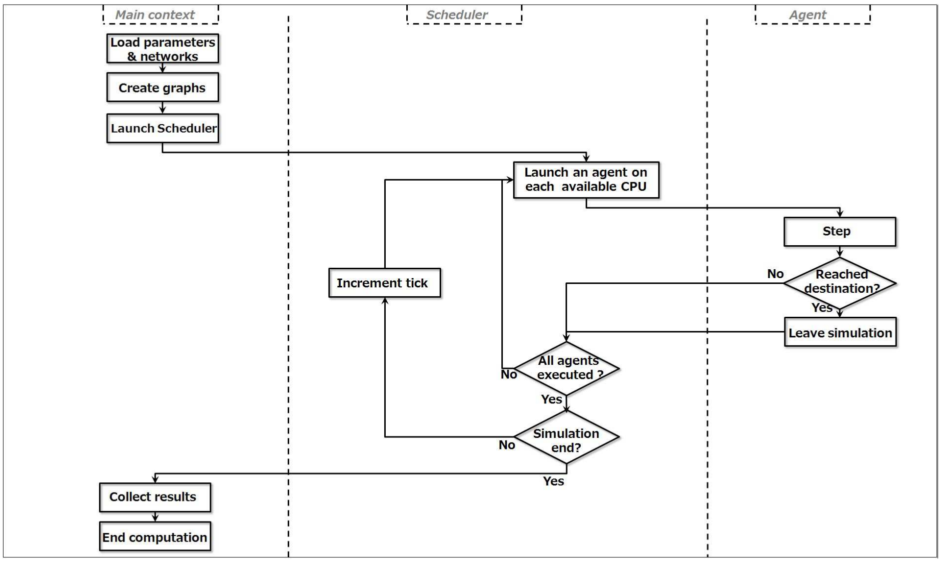

Figure 1. Workflow of the simulation

Besides, there exists several multiagent simulators for travelers mobility. For instance, Transims [9] simulates multimodal movements and evaluates impacts of policy changes in traffic or demographic characteristics. Miro [10] 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. 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 multimodal: they concern private cars, all the public transport modes as well as ridesharing and pedestrians. Passengers routes are monitored and alternatives are proposed to them if something wrong happens with their itinerary.

3. Data and Parameters

The workflow of Figure 1 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 as described in section 5 and launches the scheduler. Reapst is based on a discrete event scheduler. The default scheduler iterates over the agents and executes their *step* method. We have replaced it by a new scheduler that launches the active agents in parallel over all the available CPUs. When launched, each agent executes his step method. The behavior of each agent type is described in section 4. Finally, when the simulation duration is reached, the results are reported and the simulation ends.

3.1. Data

The input data of the simulator are: i) the road network, ii) the public transport network, iii) the transfer mapping, iv) the timetables of the public transport vehicles, v) the pedestrian network, vi) the travel patterns, vii) 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 flux (vehicles/hour), the traffic density and speeds. A public transport network is composed of transport lines, each of which 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 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.

3.2. Parameters

3.2.1. 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.

3.2.2. 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.

[3] This correspondance mission-itinerary has to be kept in mind as we will be refering to it in section 5.

- 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].

3.2.3. 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.

4. Multiagent System

4.1. 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 base their computation on the latest status of the networks. The graph representation and the paths calculations by the planner agents are described in section 5. 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.

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

4.2. 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 one 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.

4.3. Car Agents

When created, a car agent has an origin and a destination. If travel patterns exist for the considered network (cf. section 3), 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. He 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 5 for the use of these registrations). At each simulation tick, the car agent checks if he has reached his destination. If so, he leaves the simulation. If not, he keeps on travelling.

4.4. 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. All the onboard passengers are moved to the same coordinates at the same time by the vehicle. That means that, when they are onboard a vehicle or a car, traveler agents delegate the control of their movements to the vehicle agent when they decide to take him. 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.

4.5. 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).

4.6. Itinerary Monitoring

Each traveler agent and each car agent monitor their itinerary. The monitoring purpose is to ask for a new itinerary in two cases: a) 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); b) there is an event on the transport network that impacts the traveler or the car planned itinerary. To be aware of the only events that concern him, the agent subscribes 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 subscribed agents. The planning process described in the previous paragraph is then launched with the new travel times associated with the network.

5. Multimodal Itinerary Planning

The planner agent is responsible for computing the itineraries for cars and passengers. To this end, he executes the A-Star algorithm [11] on the road and multimodal networks. The value of h - 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.

5.1. 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 2). 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 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 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 planner agent interrogates the stops of the best found itinerary to infer the sequence of vehicles that the passenger agent will have to take and sends back the result to the traveler agent.

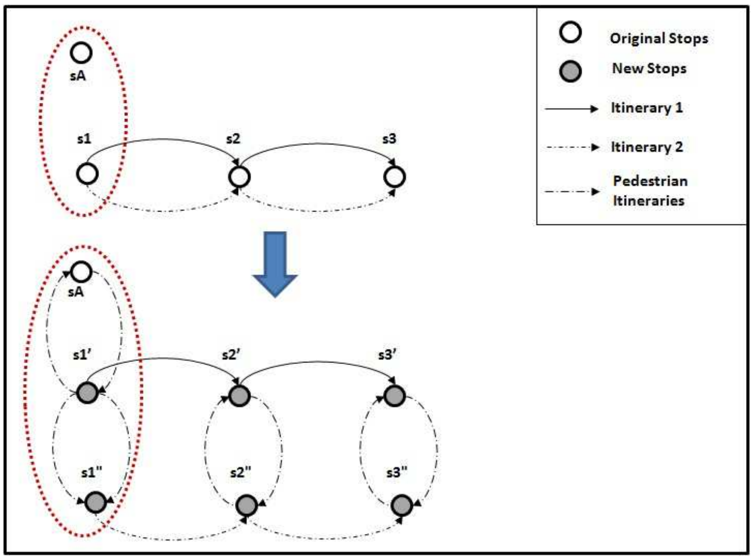

Figure 2. Multimodal Network Transformation

5.2. Ridesharing

In this paper, 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 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 planner agent 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 by a public transport itinerary. To this end, let $it_t = v_1, \ldots, v_n$ the itinerary of the traveler agent. The planner agent 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

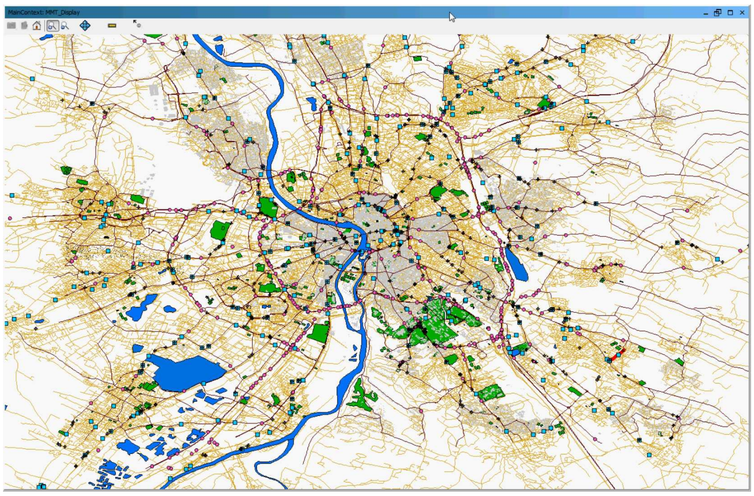

Figure 3. Simulation Execution

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.

We demonstrate the use of the platform for the city of Toulouse (cf. Figure 3), for which we have detailed data, including urban travel demand models. We have considered the main roads of the transport network of Toulouse with 13,226 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 multi-agent 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 [1].

Since the transport network parameters feed our simulator, the positions of our agents are compliant with the real life situation. In the future, we plan to implement two identical applications with two different simulators that we would compare quantitatively.

6. Conclusion and Perspectives

In this paper, we have described the main building blocks of an agent-based simulator for multimodal travelers, which are easily reproductible in other simulators pursuing similar objectives. We have focused on the main capabilities of the agents together with the data and parameters necessary for such a simulator to work properly. This application simulates individual spatiotemporal positions that are

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

compatible with the real situation. It is now possible to integrate transport applications that use these positions to provide an advanced service, such as route guidance, urban parking 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. Our second ongoing research is related to the first one. When we simulate realistic number of agents in the transport network, the problem of equilibrium arises. Indeed, if we send too many cars or travelers to the same itineraries, we risk to create congestions ourselves. We has to assign travelers and cars to itineraries while respecting their distribution aver competitive paths and not send them all to the same roads or vehicles.

References

[1] 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.

[2] 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.

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

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

[5] 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.

[6] 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.

[7] Patrick Taillandier, Duc-An Vo, Edouard Amouroux, and Alexis 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.

[8] 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.

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

[10] 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.

[11] 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.