Transportation Research Part C 45 (2014) 83–98

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Multi-agent simulation of individual mobility behavior in carpooling

Stéphane Galland [a,*], Luk Knapen [b], Ansar-Ul-Haque Yasar [b], Nicolas Gaud [a], Davy Janssens [b], Olivier Lamotte [a], Abderrafiaa Koukam [a], Geert Wets [b]

[a] Multiagent Group, IRTES-SET, UTBM, 90010 Belfort Cedex, France[1]
[b] Transportation Research Institute (IMOB), Hasselt University, Wetenschapspark 5 bus 6, 3590 Diepenbeek, Belgium[2]

ARTICLE INFO

Article history:
Received 14 May 2013
Received in revised form 8 November 2013
Accepted 31 December 2013
Available online 22 January 2014

Keywords:
Carpooling problem
Multi-agent simulation
Organizational model
Janus platform
JaSim environment model
FEATHERS

ABSTRACT

Carpooling is an emerging alternative transportation mode that is eco-friendly and sustainable as it enables commuters to save time, travel resource, reduce emission and traffic congestion. The procedure of carpooling consists of a number of steps namely; (i) create a motive to carpool, (ii) communicate this motive with other agents, (iii) negotiate a plan with the interested agents, (iv) execute the agreed plans, and (v) provide a feedback to all concerned agents. In this paper, we present a conceptual design of an agent-based model (ABM) for the carpooling a that serves as a proof of concept. Our model for the carpooling application is a computational model that is used for simulating the interactions of autonomous agents and to analyze the effects of change in factors related to the infrastructure, behavior and cost. In our carpooling application, we use agent profiles and social networks to initiate our agent communication model and then employ a route matching algorithm, and a utility function to trigger the negotiation process between agents. We developed a prototype of our agent-based carpooling application based on the work presented in this paper and carried out a validation study of our results with real data collected in Flanders, Belgium.

1. Introduction

Nowadays, carpooling is an emerging transportation mode that is eco-friendly and sustainable as it enables commuters not only to save the travel cost, such as fuel, toll and parking costs, of the carpooling participants, but also to reduce emissions and traffic congestions. Carpooling, known as ride-sharing, is the sharing of a car between people (agents) from a certain origin to a specific destination. Thus, in order to study the carpooling concept, we should take into account the interactions of two or more agents throughout the carpooling process. The procedure of carpooling consists of a number of steps, namely: (i) create a motive to carpool, (ii) communicate this motive to other agents, (iii) negotiate a plan with the interested agents, (iv) execute the agreed plans, and (v) provide a feedback to all concerned agents. Creating a motive means that a traveler (agent) may choose to carpool because of the availability of travel resources, time, monetary and route cost constraints.

Moreover, change in some socio-economic factors such as the increase in fuel price, in parking costs, or in the implementation of a new traffic policy, may trigger the initiative to carpool. Once to the decision has been made to carpool, the traveler

[1] http://www.multiagent.fr.
[2] http://www.uhasselt.be/imob.
* Corresponding author. Tel.: +33384583418; fax: +33384583342.
E-mail address: stephane.galland@utbm.fr (S. Galland).

0968-090X/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trc.2013.12.012

(agent) will try to find one or more potential partners (agents). First, each individual looks in its social network to find carpooling companions. If none can be found, global web-based matching advisors can be used. When a company plans to deploy such matching advisor software, it shall perform thorough testing because a failing service and wrong advice provision will cause customer loss. Furthermore, the company will be interested to study the transient effects occurring during the initial deployment stage. Neither this evaluation nor the testing can be performed with the help of real users. Hence, a virtual agent community is built. The agent-based model is used to evaluate the global carpooling advisory software.

The carpool initiating agent will send a request to other interested agents in its vicinity. If one or more agents who receive this request are willing to carpool, then they begin the negotiation phase. In this phase, these agents will negotiate about sharing their travel resources and optimizing total costs and daily schedules. After reaching a compromise, these agents can do carpooling. Meanwhile, an agent can appraise its partners according to their degrees of faithfulness to the carpooling. We call this degree of faithfulness the agent reputation. This reputation factor can serve as a criterion for the selection of a potential partner for carpooling. An agent-based model (ABM) is a class of computational models for simulating the actions and interactions of autonomous agents with a view to assessing their effects on the systems as a whole (Ferber, 1999; Niazi and Hussain, 2011). ABM is now widely used for modeling increasingly complex systems (Macal and North, 2005; Cossentino et al., 2010). Application of ABM is not only limited to the computer science domain. Currently, many research areas such as transportation behavior modeling, need to analyze and understand the complex phenomenon of interactions between different entities. While traditional modeling tools cannot catch the complexity, ABM can do it through modeling the interaction of autonomous agents and deducing the rules for such a system. We, therefore, in this paper propose an agent-based interaction model for the carpooling application.

This paper briefly describes a conceptual design of the carpooling application, initially proposed by Cho et al. (2012), Bellemans et al. (2012), and Galland et al. (2013). It uses an agent-based model on the Janus platform[3] (Galland et al., 2010). A simulation model needs to be created to support the individual behaviors of the participants. The contribution of this paper is the design and the implementation of an agent-based model upon the Janus multi-agent platform. This platform permits to individuals to (i) select the best transport mode according to their characteristics; (ii) maintain a social network; (iii) negotiate for carpooling; and (iv) carpool the driver and the passengers of a car.

Section 2 presents some related work to the carpooling concept and ABM. Section 3 explains our ABM for the carpooling application with details of the activities and the roles of the agents, and of the environment. Section 4 gives several implementation notes and experimental results. Section 5 is dedicated to our concluding thoughts and ideas for future work.

2. Background

Research on the carpooling concept is largely separated into two parts: (i) technical studies and (ii) empirical studies. The first ones focus on the development of carpooling support systems with techniques of travel route matching (DeLoach and Tiemann, 2012; Massaro et al., 2009). In the second part, the overall trend of carpooling – or of the interrelationship between willingness-to-carpool and the socio-economic attributes of the carpooling participants – is treated in general (Kamar and Horvitz, 2009; Horvitz et al., 2005). The previously mentioned studies are limited, and they do not consider the potential agent (participants) interactions to perform carpooling.

Most transportation-related applications of ABM are related to vehicle routing, pedestrian-flow simulation or demand modeling efforts (Bernhardt, 2007). Among these applications two of the more widely known are the ABM simulation platforms TRANSIMS and MATSIM. TRANSIMS, developed by Los Alamos Lab, is designed to supply transportation planners with more delicate information about traffic impacts, energy consumption, land-use planning and emergency evacuation (Smith et al., 1995). MATSIM is also a large-scale agent-based simulator similar to TRANSIMS, but it is different using of XML and quickly run simulation, due to a simplified traffic simulator (Waraich et al., 2009). Those applications only consider the whole effect of each agent's action on a system, and cannot handle a detailed agent-to-agent or agent-to-environment coordination, communication and negotiation.

According to Odell et al. (2002), "the environment provides the conditions under which an entity (agent or object) exists." The author distinguishes between the *physical environment* and the *communication environment*. The physical environment provides the laws, rules, constraints and policies that govern and support the physical existence of agents and entities. The communication environment provides (i) the principles and processes that govern and support exchanges of ideas, knowledge and information, and (ii) the functions and structures that are commonly deployed to exchange communication, such as roles, groups and interactions protocols between roles and groups.

Odell et al. (2002) define an agent's social environment as "a communication environment in which the agents interact in a coordinated manner." This approach is shared by Ferber et al. (2006), Cossentino et al. (2010), and Galland et al. (2009), who proposed to integrate the environment with organizational models. The Janus platform (Gaud et al., 2008; Galland et al., 2010) provides an implementation of the agent-based concepts, and of the Capacity–Role–Interaction–Organization metamodel (Cossentino et al., 2010). The JaSim library[4] (Galland et al., 2009) provides a model of the physical environment upon the janus platform. In the rest of this paper, the graphical notation is inspired by Cossentino et al. (2010).

[3] http://www.janus-project.org.
[4] http://www.multiagent.fr/Jasim_Platform.

3. Carpooling model

The carpooling model is designed to consider the individual behaviors during the carpooling process. This process is composed by the five steps described in Section 1. The goal is to simulate how everyone is deciding to carpool, and how the carpooling process is executed. From the simulation's results, it is possible to understand the causes why people are carpooling in a given area.

An agent is defined as someone who lives in our study area and executes his or her own daily schedule in order to satisfy his or her needs. A schedule is a combination of a number of trips associated with a number of activities. An agent is a member of household such as the husband, the wife, the parents or the children. (S)He also is a member of the society such as a friend, a colleague, a neighbor, an employee (or an employer) or a student. In our model, we consider the socio-economic attributes, including age, gender, income, education, relationship (within a family), job, vehicle and driving license ownership, as a set of input data.

These schedules and attributes are supplied by FEATHERS (Bellemans et al., 2010), an *activity-based traffic-demand model*, which is developed by IMOB – Hasselt University Belgium. Activity-based modeling (ActBM) is a technique that predicts the daily travel agenda (schedule) for each member of a synthetic population. Most ActBM generate predictions for a single day. For each predicted activity, the ActBM specifies the activity type, start time, duration, location as well as the duration and transportation mode for the trip to reach the activity location. Activity-based models are micro simulators: the behaviors of the individuals are simulated. This allows investigating the overall effects of traffic demand management policies. FEATHERS is an operational activity-based model for the region of Flanders (Belgium); it generates the schedules for a given day of the week. The input data from FEATHERS consists of:

- the synthetic population for the study area. This contains socio-economic data (household composition, education level, income category, age category, etc.) describing each individual so that the distributions fit the census data,
- an area subdivision into traffic analysis zones (TAZ),
- land-use data for each TAZ. This consists of tens of attributes including number of people living in the TAZ for several age and employment categories, amount of people employed in the TAZ in several economic segments (industry, agriculture, education, distribution, hospitals, etc.),
- impedance matrices specifying the travel time and distance between TAZ for off-peak, morning-peak and evening-peak periods and for several transportation modes (car, slow, public transport),
- a set of decision trees trained using large scale (periodic) travel surveys. Those data essentially specify individual behavior as a function of socio-economic data and partial schedule characteristics.

A schematic overview of the data flows in FEATHERS is given by Fig. 1. The Flemish model is characterized by the following data:

- *Synthetic population size:* 6 million people.
- *Number of TAZ:* 2368.
- *TAZ area (average value):* ≈5 km^2.
- *Number of diaries in survey:* ≈8000.

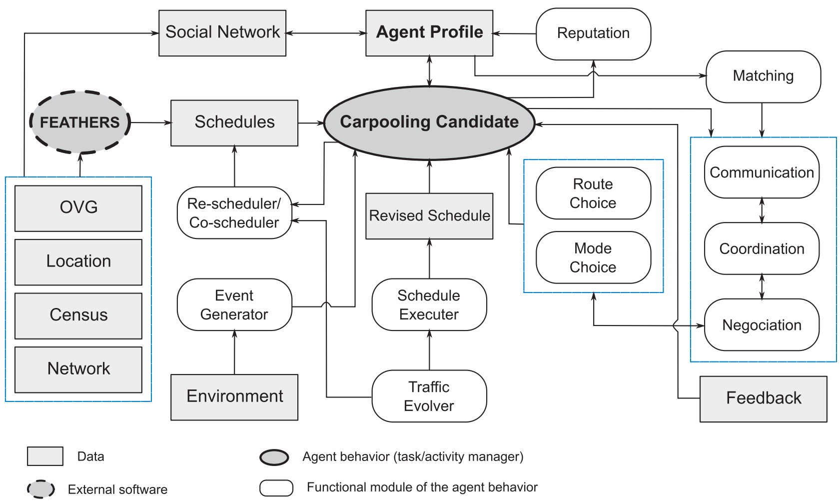

Fig. 1. Agent-based Model for the Carpooling Application. The central node shows the individual planning to carpool. The left hand part shows the initial daily plan generation by the FEATHERS activity-based model (see also Fig. 2).

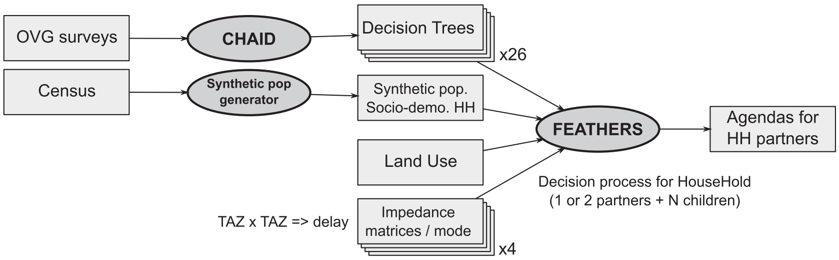

Fig. 2. FEATHERS Data Flows and Problem Size Specification for the Flanders Region. The *CHAID* process generates a set of decision trees from survey data. *SynPopGen* creates a synthetic population within the study area. FEATHERS predicts a daily agenda for each synthetic individual.

Fig. 2 shows a conceptual overview of FEATHERS in a data flow diagram. OVG[5] surveys results are used to establish a set of 26 decision trees. Training is done using the CHAID (CHi-squared Automatic Interaction Detection) technique (Kass, 1980). The decision process used by the individuals is modeled by applying the decision trees in an order. The population within the study area is generated by *SynPopGen*. That population mimics the real population; a large set of marginal distributions of the real and synthetic population are closely similar. Land-use data consist of the number of jobs, the number of inhabitants, the number and the size of the schools, and many other indicators for each of the traffic analysis zones (TAZ). Those data are used to calculate the attraction for each TAZ. Finally, the impedance square matrices give the travel times between TAZ for each hour of the day; those are calculated from the network by loading the road network with the expected traffic.

Agents follow a number of steps, including the goal setting, the scheduling based on a given resource and environment, and the execution of their schedule. These steps may be modeled within an activity diagram, which is shown in Fig. 3. Six major activities are considered:

1. "Mode selection:" the agent is selecting its preferred transport mode.
2. "Matching:" the agent is selecting its partners.
3. "Negotiation:" the agent is negotiating with its partners for the details of the carpooling.
4. "Driving:" each driving agent is simulated on a road network.
5. "Feedbacks:" each agent computes the feedback at the end of the day according to the activities of the day.
6. "Environment Updating:" each non-carpooling agent registers its mobility behavior in the environment.

The following sections detail the global behavior of an agent and these major activities.

3.1. Mode selection activity

The agent is selecting its preferred transport mode. Here, the FEATHERS data are used to determine the mode choice stochastically. If the agent decides to carpool, he goes in Activity 2, otherwise he runs its activity in Activity 6.

3.2. Matching activity

The matching is applied in both *local* and *global* exploration phases. In both cases, matching precedes the negotiation phase where final decisions to carpool are taken. A person looks for other individuals to cooperate while executing its periodic trip (*periodicTripEx*): this is called *exploration*. *Local* exploration within the private social network (*PrivNet*) is applied before the *global* exploration. *PrivNet* is represented by an organization (Fig. 4) in which each agent is playing a role, and has a relationship with the other members of the organization. If *carpool candidates* can be found within an agent's *PrivNet*, they will be contacted first (as preferred candidates). *Global* exploration is applied only in a second stage when no suitable *pool* was found in *PrivNet*. In the *global* exploration phase, the *matcher* provides advice about which *pool*s an individual should negotiate with. This corresponds to the use of an online service by which to explore the set of formerly unknown carpooling candidates. Registration in this service implies first posting some descriptive characteristics such as age, gender, education level, special interests (e.g. music style preferences), job category, and driver license availability. Those qualifiers are used because it is known that continued successful cooperation between people requires a minimal level of similarity.

Two people may do carpooling together if their $CP = \{L, SR, I, R\}$ are matching. Location (L) is matching the start location of the agents. Spatial Relevance (SR) is the match between the paths from the origin to the destination of all interacting agents. Interests (I) and Requirements (R) are matching the interests and the requirements of each agent in their profiles, respectively. These matchings are based upon the similarity models described in the following sections.

[5] OVG: "*Onderzoek Verkeers Gedrag*" means Travel Behaviour Research in Dutch.

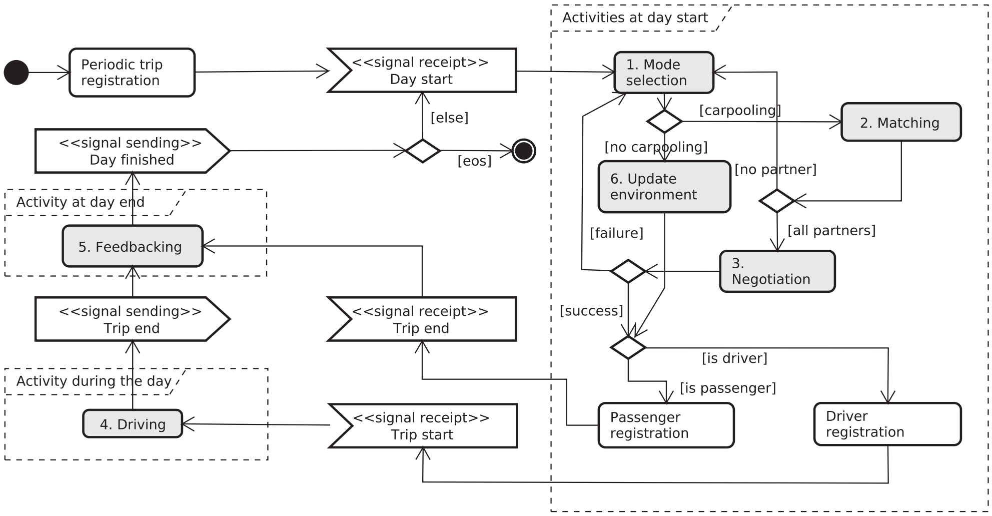

Fig. 3. Diagram of the activities of a carpooling agent. The numbered activities (with a gray background) are the major activities described in details in the text. The others are secondary activities related to the implementation on the JANUS platform.

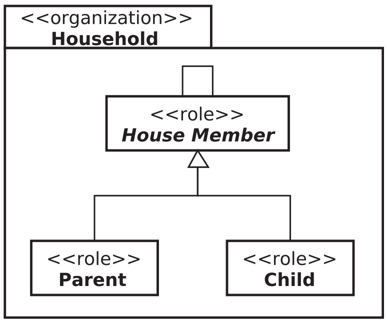

Fig. 4. Household.

3.2.1. Profile similarity

The candidate carpooler specifies a set of N_A attribute values: those constitute the candidate's *profile*. The model uses the *similarity* between two profiles as a predictor (one independent variable) for the *logit* model. The distance between two attribute tuples a_0 and a_1 having N_{OA} ordinal attributes is the Euclidean distance divided by a scale factor to normalize the distance (map to interval $[0, 1]$) as described by Eq. (1).

$$d(a_0, a_1) = \sqrt{\frac{\sum_{i \in [1, N_{OA}} (a_0[i] - a_1[i])^2}{N_{OA}}} \tag{1}$$

Continuous variables are combined into a single distance value d_C, and discrete ordinal values are combined into another one d_D. The range of d_D is a finite subset of $[0, 1]$. The similarity values $s_C = (1 - d_C), s_D = (1 - d_D)$ and $s_E = (1 - d_E)$ are used as independent variables for the *logit* estimator.

3.2.2. Path similarity

The *Global CarPooling Matching Service (GCPMS)* per hypothesis has no information about carpool parking, potentially being used (because that is not specified by the candidates). Therefore, it is assumed that people board and alight at home and work locations only. The periodic trip executions need to be matched, not people. A periodic trip on Wednesday from A to be leaving at about 08:30 h needs to be matched with an another one having similar characteristics. Of course, the people involved shall be mutually compatible but they are not the primary subject of matching. A particular individual can periodically carpool with several people for different trips in the week (on Monday with colleague A, on Tuesday with neighbor B who differs from A). The owner of the first *Periodic Trip Execution*, abbreviated by *periodicTripEx*, is the driver.

Let:

- O_i, D_i: denote respectively the *origin* and *destination* locations for individual i (e.g. home and work locations);
- $r(a,b,t)$: denote the route from a to b when starting at time t that is optimal with respect to some cost function $c(r)$ based on distance and travel time;
- $d(r,t)$: denote the duration to travel the route r starting at time t;
- $l(r,t)$: denote the length to travel the route r starting at time t;
- $c(r)$: denote a cost function based on route length $l(r,t)$ and route travel duration $d(r,t)$;
- $p_{i,solo}(O_i, D_i, t)$: denote the optimal path from O_i to D_i when individual i drives alone (solo) and starts at time t;
- $\overline{p}_{i,solo}(O_i, D_i, t)$] denote the optimal path from O_i to D_i when individual i drives alone (solo) and ends at time t;
- $p_{i,carpool}(O_i, D_i, t)$: denote the optimal path from O_i to D_i when individual i drives the carpool trip via O_j and D_j for $i \neq j$ and starts at time t;
- $\overline{p}_{i,carpool}(O_i, D_i, t)$: denote the optimal path from O_i to D_i when individual i drives the carpool trip via O_j and D_j for $i \neq j$ and ends at time t;
- $pathSim_d()$: denote the path similarity function for the case where the earliest departure is given;
- $pathSim_a()$: denote the path similarity function for the case where the latest arrival is given.

The ratio between the lengths of the optimal routes for the driver is used as a *path similarity function*. For the *given earliest departure* case (starting at t_0) where A is the driver, and the trip is $O_A \rightarrow O_B \rightarrow D_B \rightarrow D_A$, this leads to Eq. (4).

$$t_1 = t_0 + d(r(O_A, O_B, t_0)) \tag{2}$$
$$t_2 = t_1 + d(r(D_B, D_A, t_1)) \tag{3}$$

$$pathSim_d(pte_A, pte_B, c()) = \frac{c(O_A, D_A, t_0)}{c(O_A, O_B, t_0) + c(O_B, D_B, t_1) + c(D_B, D_A, t_2)} \tag{4}$$

Note that t_1 denotes the time at which the carpool trip leaves O_Bn and t_2 denotes the time at which the carpool trip leaves D_B. Moreover, the Inequalities 5 and 6 hold since the departure times can differ.

$$pathSim_d(pte_A, pte_B, c()) \neq pathSim_a(pte_A, pte_B, c()) \tag{5}$$
$$pathSim_a(pte_A, pte_B, c()) \neq pathSim_a(pte_B, pte_A, c()) \tag{6}$$

The departure time can have a large effect upon the trip duration. In the first GCPMS, this dependency is ignored due to lack of data. Because of the availability of speed profiles registered using GPS navigators, it will become feasible to take the time dependency into account (which will lead to more accurate negotiation outcome prediction) in the near future although that will require a large amount of data pre-processing and data storage. By ignoring time dependency, Eq. (4) is reduced to obtain the Eq. (7).

$$pathSim_d(pte_A, pte_B, c()) = \frac{c(O_A, D_A)}{c(O_A, O_B) + c(O_B, D_B) + c(D_B, D_A)} \tag{7}$$

3.2.3. Time interval similarity evaluation for matching

It is not feasible to ask the individuals to register the piecewise linear preference function mentioned in Section 3.3.2. People are assumed to be prepared to register simply a time interval only. Hence the *preference* value is assumed to be a constant f over the time interval specified.

The negotiation outcome is assumed to be positively correlated to the intersection's length of the intervals associated with the *periodicTripExs* to compare. The time interval similarity *tis* is given by the equations:

$$t_0 = \max(t_{i_A,0}, t_{B,0}) \tag{8}$$
$$t_1 = \min(t_{i_A,1}, t_{B,1}) \tag{9}$$
$$tis(i_A, i_B) = t_1 - t_0 \tag{10}$$

For a given pair of *periodicTripExs*, *tis* values are fed into the *logit* estimator as two independent variables; combining them into a single value would cause a loss of information.

3.2.4. Reputation of the agents

Each driver has a *safety reputation* value (*sReputation*) that evolves over time due to the qualifications by the passengers. They are the individuals who participated in an agreement in which the person being evaluated is the driver.

Each person (both drivers and passengers) has a *timeliness reputation* (*tReputation*). This qualifies the individual as being on time to start the trip (not wasting someone else's time).

Both reputation values are handled in the same way. Details are explained here for the *sReputation* case. The most-recent qualification is saved by each issuer. The *sReputation* is calculated as a weighted average of the values in the list of all the qualifications: the weight decreases with the age of the qualification, and increases with the duration of the cooperation $a.dur(n.ts())$ up to the moment of the qualification (the agreement lifetime). Let $\mathcal{Q}_{i_0}$ be the qualifications list for individual

i_0. Let $a_j^{i_0}$ denote an agreement in which individuals i_0 and j cooperated. Let $a.iss()$ denote issuer of the qualification. $n.ts()$ denotes the time at which the qualification was issued and $a.dur(t)$ denotes the lifetime of agreement at a time t. Then

$$age = now - n.ts() \tag{11}$$

$$w^{i_0}_{n.iss()} = \exp(-\alpha \cdot age) \cdot a^{i_0}_{n.iss()}.dur(n.ts()) \tag{12}$$

$$sReputation_{i_0} = \frac{\sum_{n \in \mathcal{Q}_{i_0}} n.sRep() \cdot w^{i_0}_{n.iss()}}{\sum_{n \in \mathcal{Q}_{i_0}} w^{i_0}_{n.iss()}} \tag{13}$$

Every agent keeps a list containing a perceived safety reputation value for a limited set of other agents. In the exerciser, a specific agent can be qualified by zero or more safety reputation values (each one of which is owned by a peer). Every agent can determine the reputation of another agent using a method that is not specified in this section and overriding the value already in place (if any). Furthermore, everyone can adjust the reputation of peers based on *gossip* as follows. At a random moment in time, an emitter agent a_e can multicast its reputation value $R^e(q)$ to qualify agent a_q to a subset of agents directly connected to it in the social network. The *r*eceiver *ar*:

1. retransmits the reputation message with a given probability p_r (hence simply drops it with probability $(1 - p_r)$),
2. adjusts its own perception of a_q with a given probability p_a (hence simply ignores it with probability $(1 - p_a)$).

Consider agents a_e, a_q, ar, a_v that are all pair wise different. a_e emits a qualification about a_q that reaches a_r via its neighbor a_v. If a_r has not yet registered an opinion about *aq*, the value for $R^r(q)$ is 0. Reputation update by receiver a_a is done by:

$$\alpha = 2^{-d(e,r)} \tag{14}$$

$$\beta_{r,v} \in [0, 1] \tag{15}$$

$$R_q^{\prime r} \leftarrow \frac{R^r(q) + \alpha \cdot \beta \cdot R^e(q)}{1 + \alpha \cdot \beta_{r,v}} \tag{16}$$

where $d(e,r)$ is the distance between emitter and receiver in the network and $\beta_{r,v}$ is the strength of the link between a_r and a_v.

3.3. Negotiation activity

The negotiation is the process during which the members of a *pool* are negotiating the details of their *periodicTripEx*s (time window, who is the driver and the passengers).

3.3.1. Overview of the negotiation activity

If the negotiation fails, the agent goes back to Activity 1; otherwise to Activity 4. The negotiation is based on a specific protocol. According to our organizational approach, the agents who are negotiating together are members of the same organization "Negotiation Pool" (Fig. 5). The negotiation protocol is described as a sequence of messages exchanged by the different participants, as illustrated by Fig. 7. This negotiation protocol is re-launched when new time windows are proposed, and until all the participants were found (success of the negotiation), or when the initiator of the negotiation is considering that it is impossible to obtain a common decision among the participants (failure of the negotiation).

If an individual joins a *pool*, (s)he is added to the *PrivNet* for all other participants in the *pool* (if still required), so that if i_0 and i_1 cooperate in a *pool*, (i_0, i_1) and (i_1, i_0) belong to each other's private networks. Because the links are never removed from the *PrivNet*, if i_0 and i_1 ever carpooled, $(i_1) \in PrivNet(i_0, 1) \wedge (i_0) \in PrivNet(i_1, 1)$. Candidates register, join and leave the *pools* at random moments in time. As a consequence, the main data structures dynamically change due to events external to the matching process.

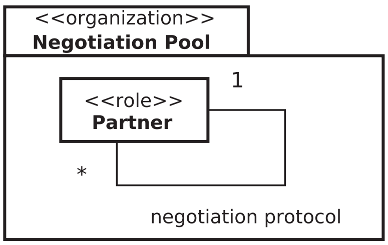

Fig. 5. Negotiation pool.

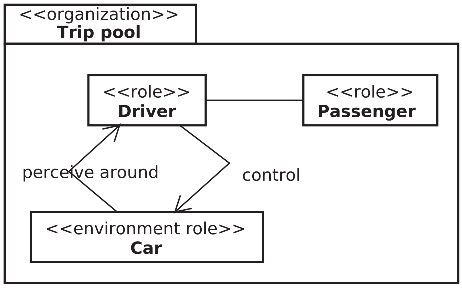

Fig. 6. Trip pool.

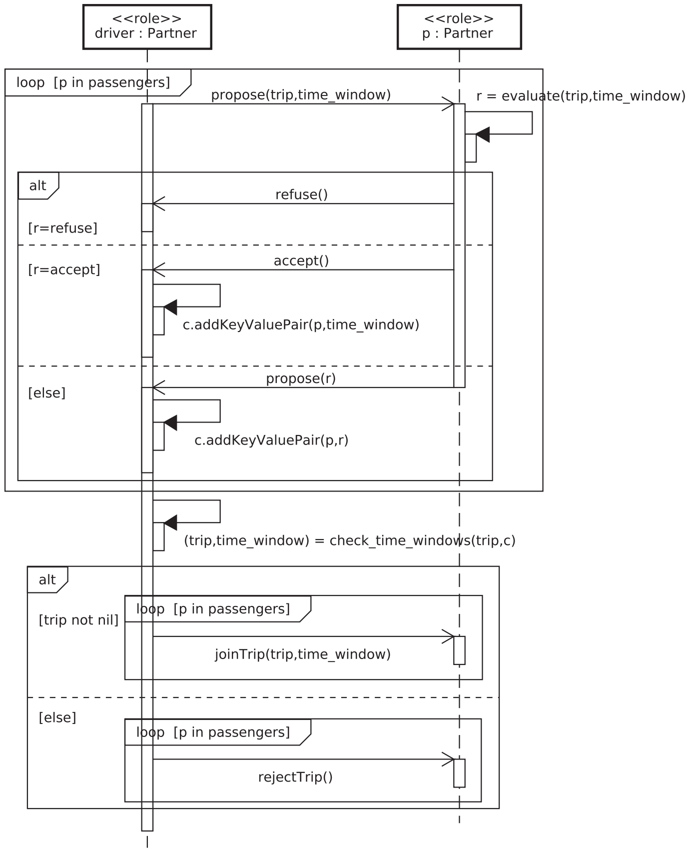

Fig. 7. Sequence diagram of the negotiation protocol.

The negotiation process is a discrete variable with the outcome value *success* (yes) or *failure* (no). A *logit* model is used to predict the negotiation outcome. Negotiation results fed back to the *Global CarPooling Matching Service (GCPMS)* are used to determine the coefficients in the *logit* model by linear regression.

After having found a good match, the matcher conveys its advice to the involved candidates (the owners of the matched *periodicTripEx*). They evaluate the proposal, negotiate about carpooling and possibly agree to cooperate. It is impossible to predict the negotiation phase with certainty; reasons are:

1. Negotiation covers the driver's selection, the co-route determination, and the re-scheduling (daily planning adaptation) for the cooperators. Schedule adaptation makes use of *VOT* (individual specific *Value Of Time*). An advisory mechanism does not have all required data available nor has any knowledge of the private goals (and in general, the beliefs, desires, intentions – *BDI*) of the individuals (agents) involved in a negotiation.
2. The total distance driven cannot be predicted by the *matcher* when carpool parking is involved, because in such a case, the co-route can be tree structured. Hence the path similarity function delivers only an approximation of the one involved in negotiation.

3. People are assumed to be prepared posting a minimal amount of data about the time intervals that suit them for departure an arrival respectively; candidates are supposed to specify just the interval boundaries. However, during negotiation, they can make use of *preferences* to state that one of a set of proposed intervals suits better than another one. Hence, the *trip times interval similarity* function available to the matcher is only an approximation for the one used during negotiation.
4. The *Cotravel_Refused* allows individuals to avoid unconditionally any advice to carpool with specific people. For privacy reasons, it is impossible for a refused individual to know the refusing party.

Therefore, the candidates convey the negotiation result back to the matcher service. This paper assumes that sufficient (financial) incentives are in place in order to make this happen. The feedback is used by a learning mechanism incorporated into the matching service. After receiving the feedback, the matching service disposes of the *periodicTripEx* and the individuals' characteristics as well as of the negotiation result; those are used to train a predictor.

3.3.2. Time interval based functions for negotiation

The *departure* (resp. *arrival*) interval for a trip (*periodicTripEx*) is the time interval that suits the traveler to start (resp. end) the trip. Let $pte.i_d()$ and $pte.i_a()$ denote respectively the *departure* and *arrival* intervals of the *periodicTripEx pte*.

The *preference* of the individual p_0 for a given moment in time is provided by the function $f_{p_0} : \mathbb{R} \Rightarrow \mathbb{R} : t \mapsto f_{p_0}(t) \in [0,1]$. The function is not required to be differentiable or continuous, but it shall be integrable. For each moment in time belonging to the departure and arrival intervals, the *preference* value needs to be specified.

The *combined preference* function is the product of the preference functions associated with two *periodicTripEx*s. It is essential to the negotiation process.

The integral of the combined preference over a fixed time interval in $[0,\infty)$ is called the *time interval suitability*. The length of the interval has a pre-specified constant C value. A suitable choice is the expected duration of the trip interruption to get someone in or outside of the vehicle. The *time interval suitability* is denoted by $S(C, i_A, f_A, i_B, f_B)$, where $i_A = [t_{i_A,0}, t_{i_A,1}]$ and $i_B = [t_{i_B,0}, t_{i_B,1}]$ are the intervals specified by the individuals A and B; f_A and f_B are the associated preference functions. The suitability function is given by the following equations:

$$t_0 = \max(t_{i_A,0}, t_{i_B,0}) \tag{17}$$

$$t_1 = \min(t_{i_A,1}, t_{i_B,1}) \tag{18}$$

$$S(t; C, i_A, i_B, f_A, f_B) = \begin{cases} \int_t^{t+C} f_A(x) \cdot f_B(x) dx & \text{if } t \in [t_0, t_1 - C] \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where t denotes the start of the boarding/alighting operation. The dimension of the *combined time interval suitability* value is $[prefUnit^2, timeUnit]$. In this context, preference is assumed to be dimensionless hence the suitability dimension reduces to $[timeUnit]$. During negotiation, $S(t; C, i_A, i_B, f_A, f_B)$ is with other functions to find a suitable time to board/alight.

Piecewise linear functions are used because they are flexible. They can easily be specified by the user (responsible for the configuration of the agent-based model), and their integration is computationally cheap. The left hand part in Fig. 8 shows piecewise linear *preference* functions, their product and the associated time interval suitability (crosshatched area under the product function). The right hand part shows the case for the same intervals where the preference function is assumed to equal one everywhere: this is the assumption made by the matching service due to lack of information: the user only specifies the boundaries for the departure and arrival intervals.

3.3.3. Agreement attributes

In this section we will present the three basic attributes important for constituting an agreement between agents.

- *Cohesion of an Agreement*; *Cohesion* is supposed to be a monotonically decreasing function of the time t, which is elapsed since the creation of the *agreement*. *Cohesion* is a monotonically decreasing function of the *pool* size s (large *pools* are more likely to disintegrate). The cohesion does not depend on mutual evaluation of carpoolers. The cohesion and the reputation shall be independent concepts. Indeed, they are fed into a probability estimator. The *cohesion value* is given by:

$$c = e^{\alpha \cdot t} \cdot e^{\beta \cdot (s-1)} \tag{20}$$

In the pair wise case, when considering a specific edge, exactly two *cohesion values* apply (one for each of the vertices (*periodicTripExs*)). Each of the cohesion values possibly applies to an *agreement*. For a given pair of *periodicTripExs* (pte_0, pte_1), the c_0 and c_1 can relate to either different *agreements* or to a single one. The meaning of the tuple (c_0, c_1) depends on the number of *agreements* involved. Therefore, *cohesion values* are combined into a single *cohesion based indicator* using the function given in Eq. (22). The second case in Eq. (22) is for when both *periodicTripEx*s are members of an *agreement* (but not necessarily to the same one). Let $pte_0, pte_1 \in \mathcal{T}$ the *periodicTripEx*s involved. Let c_0 and c_1 denote the respective corresponding *cohesion values*, and $p.T()$ denotes the list of the *periodicTripEx*s that are involved in the pool p. Let $pte.a()$ denotes the *agreement* covering *pte* when it belongs to a *pool*. The *cohesion indicator* $\overline{c}(pte_0, pte_1)$ is a measure for the cohesion between two *periodicTripEx*s, when they already form a pair, and for the feasibility to get them released when they are bound in pairs with others.

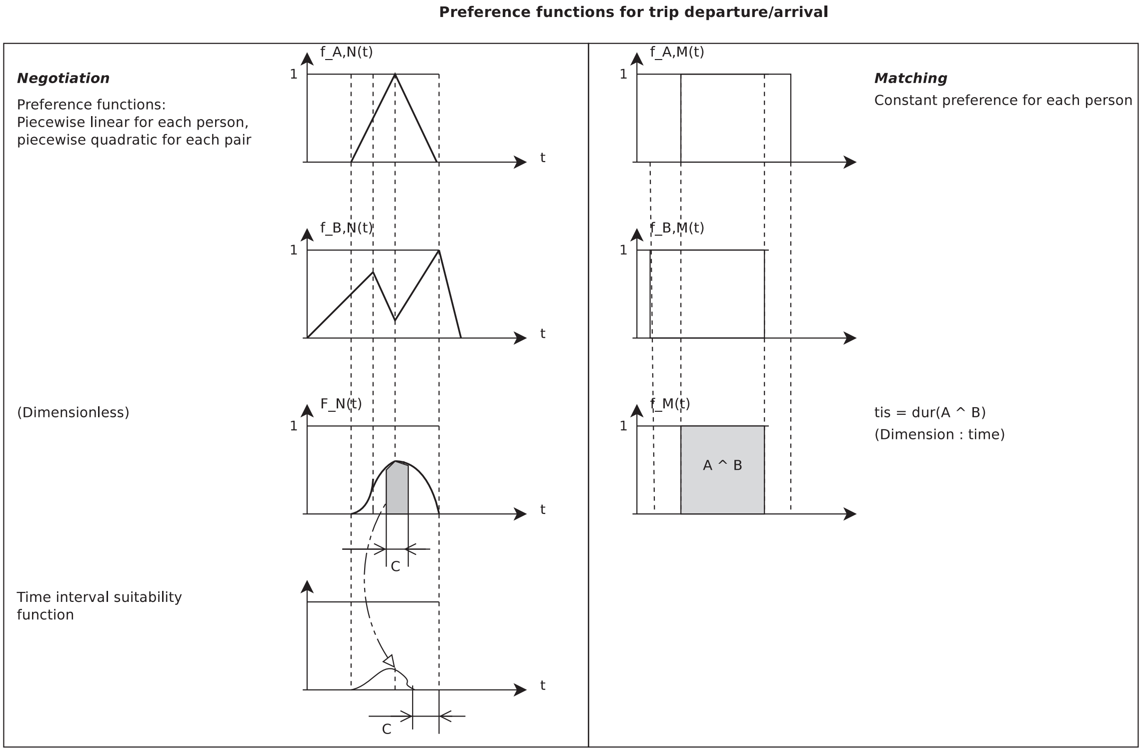

Fig. 8. (Left) Time similarity used while negotiating: $f_{A,N}(t)$ and $f_{B,N}(t)$ are time preference functions for specific intervals. $F_N(t)$ is the *combined preference* and the size of the cross-hatched area is the resulting *time interval suitability function*. (Right) Time similarity used by the matcher: all preference functions equal 1 because users are expected to only submit feasible time intervals.

$$pte_x.a() = \begin{cases} nil & \text{if } \nexists p \in \mathcal{P} | pte_x \in p.T() \\ p.a() & \text{if } \exists p \in \mathcal{P} | pte_x \in p.T() \end{cases} \tag{21}$$

$$c_x = \begin{cases} 0 & \text{if } pte_x.a() = nil \\ pte_x.a().c() & \text{else} \end{cases} \tag{22}$$

$$\bar{c} = \begin{cases} (1 - c_0) * (1 - c_1) & \text{if } pte_0.a() \neq pte_1.a() \\ c_0 * c_1 & \text{if } pte_0.a() = pte_1.a() \neq nil \end{cases} \tag{23}$$

In the case that both *periodicTripEx*s belong to the same *agreement*, the cohesion values are taken from that *agreement*, and in fact $c_0 = c_1$. In the other case (which also covers the case where at least one of the *periodicTripEx* is not covered by an *agreement*), the complement of the *cohesion* values is used. When neither of the *periodicTripEx* belongs to an *agreement*, $\bar{c} = 1$.

- *Safety Reputation*: While evaluating the success probability for a *pool*, exactly one *sReputation* value applies since only the *sReputation* for the driver is relevant.
- *Timeliness Reputation*; The *tReputation* of an individual i_0 applies to an *agreement*, and exists only exists as long as the agreement holds. It can only be affected by the partners in the agreement that are different from i_0 (in the pair wise case, there is only one such partner). The *tReputation* is an evaluation score assigned by the partners.

3.4. Driving activity

The driving activity corresponds to the execution of the trip. The driver controls its car (with the carpooled passengers inside) on the roads. The road network is represented by a graph built from geographic data. The JANUS platform provides an environment model able to support the displacements of the cars on the roads (Galland et al., 2009). Fig. 6 presents the organization that is supporting the trip simulation. All the agents in a trip pool must play a role in an instance of this organization. The behavior of the Driver role is composed of two layers: (i) the path planning on the roads and (ii) the path following and collision avoidance.

Algorithm 1. Behavior of the driving agents

1: **function** DRIVER BEHAVIOR
2: **repeat**
3: $path \leftarrow \{\langle s\rangle.e | \forall s' \in b, path = b.\langle s\rangle.e \wedge position \in s \wedge position \notin s'\}$
4: **if** $\neg is\ FreePathpath$ **then**
5: $g \leftarrow firstJunctionInpath$
6: **if** g **then**
7: $p \leftarrow \mathrm{astar}position, g$
8: $path \leftarrow \{\langle s\rangle.e | \forall s' \in b, path = b.\langle s\rangle.e \wedge g \in s \wedge g \notin s'\}$
9: $path \leftarrow p.path$
10: **else**
11: $path \leftarrow astarposition, goal$
12: **end if**
13: **end if**
14: $p \leftarrow getPerceivedObjectspath, roads$
15: $update\ AttributesAccordingTop$
16: $o \leftarrow \{a | \forall b \in p, distanceposition, b \geqslant distanceposition, a\}$
17: **if** o
18: $a \leftarrow \sigma_a(followerDrivingo, acceleration, speed, position)$
19: **else**
20: $a \leftarrow \sigma_a(freeDrivingacceleration, speed, position)$
21: **end if**
22: $speed \leftarrow \sigma_s(speed + \frac{a.\Delta t}{2})$
23: $position \leftarrow position + speed.\Delta t$
24: **if** $speed > 0$ **then**
25: $waitingTime \leftarrow 0$
26: **else**
27: $waitingTime \leftarrow waitingTime + \Delta t$
28: **end if**
29: **until** $position = goal$
30: **end function**
31: **function** ISFREEPATH($path$)
32: **return** $path \neq \langle\rangle \wedge \neg path[1].isBlocked \wedge waitingTime < timeout$
33: **end function**
34: **function** FIRSTJUNCTIONIN($path$)
35: $r \leftarrow \mathrm{nil}$
36: **for all** $\langle a, b\rangle \in path \wedge \neg r$
37: $c \leftarrow \{i | i \in a \wedge i \in b\}$
38: **if** $\|\mathrm{neighbor}(c)\| \geqslant 3$ **then**
39: $r \leftarrow c$
40: **end if**
41: **end for**
42: **return** r
43: **end function**

The behavior of the agents, who are driving during the simulation, is described by Algorithm 1. Every agent maintains a sequence of connected road segments[6] to follow on the road network, says the *path*:

$$\begin{aligned} path = \langle s_i | i \in [0; n) \wedge \\ s_i \cap s_{i-1} \neq \emptyset \Rightarrow i > 0 \wedge \\ s_i \cap s_{i+1} \neq \emptyset \Rightarrow i < n\rangle \end{aligned} \quad (24)$$

Initially, the *path* is the shortest path between the position of the agent and its goal (given by the function ASTAR in Algorithm 2). At every simulation-time step, the roads already traveled are removed from the *path* (line 3 of Algorithm 1).[7]

[6] $\langle\alpha\rangle$ is the sequence of road segments such that, $\forall a \in \alpha, a$ is a road segment.
[7] $a.b$ is the sequence of road segments starting by the sequence a and finishing by the sequence b.

Algorithm 2. The A* algorithm

```
1: function ASTAR(start, goal)
2:   closeset ← ∅
3:   openset ← {start}
4:   came_from[start] ← nil
5:   g[start] ← 0
6:   f[start] ← g[start] + h(start, goal)
7:   while openset do
8:     c ← {s|∀(m, s) ∈ openset², f[s] ⩽ f[m]}
9:     if c = goal
10:      p ← ⟨⟩
11:      while goal do
12:        p ← ⟨segmentcame_from[goal], goal⟩.p
13:        goal ← came_from[goal]
14:      end while
15:      return p
16:    end if
17:    openset ← openset − {c}
18:    closeset ← closeset ∪ {c}
19:    for all n ∈ neightbor(c) do
20:      ng ← g[c] + distance(c, n)
21:      if n ∉ openset ∨ ng < g[n] then
22:        came_from[n] ← c
23:        g[n] ← ng
24:        f[n] ← g[n] + h(n, goal)
25:        openset ← openset ∪ {n}
26:      end if
27:    end for
28:  end while
29:  return ⟨⟩
30: end function
```

The path planning is dynamic: the driver adapts its path according to its perceptions from the environment jams, road-works with a variant of the A* search algorithm (Dechter and Pearl, 1985; Delling et al., 2009), with its principles closed to the D*-Lite algorithm (Koenig and Likhachev, 2005). This family of path-planning algorithms has two advantages: it enables path re-computation during the simulation according to a new state of the environment; and it is suitable for a partial knowledge of the environment's state. Lines 4–13 provide the behavior for the dynamic re-planning of the *path*. The function ISFREEPATH determines if the following segments along the *path* are traversable. If they are not, the segments to the subsequent junction along the path are replaced by the shortest path from the current position to this junction. If there is no junction on the *path*, it becomes the shortest path to the goal. The A* search algorithm is used to compute all the shortest paths. It is described in Algorithm 2. A* uses a best-first search and finds a least-cost path from a given initial road to one goal segment. As A* traverses the graph of the roads, it follows a path of the lowest expected total cost or distance, keeping a sorted priority queue of alternate path segments along the way. It uses two base functions: (i) the past path-cost function $g(x)$, which is the known distance from the starting segment to the current segment x, and (ii) a future path-cost function $h(x)$, which is an estimation of the distance from x to the goal. The function $h(x)$ must be an admissible heuristic; that is, it must not overestimate the distance toward the goal. Thus, for an application like routing, $h(x)$ might represent the straight-line distance to the goal, since that is physically the smallest possible distance between any two points or nodes.

After the driving agent has updated its path, he follows it, and he avoids collisions with the other cars around. The function GETPERCEIVEDOBJECTS at line 14 is provided by the JASIM library, and it replies the set of objects in the field-of-perception of the driver. A standard Intelligent Driver Model (Treiber et al., 2000) is used at line 18 to adapt the velocity of the car according to the ahead vehicles and to the road signs: the traffic lights and stop signs are assimilated to immobile vehicles until the driver decided to pass through. The other road signs are used to update the variables in the driving model (desired velocity). The acceleration is defined by Eq. (25), where Δv is the difference between the velocities of the agent and the ahead object. Δp is the distance to this ahead object. b is the comfortable braking deceleration. s is the security distance, and w is the desired time headway to the vehicle in front.

$$\text{followerDriving} = \begin{cases} -\frac{(v\Delta v)^2}{4b\Delta p^2} & \text{if the ahead object is far} \\ -a\frac{(s+vw)^2}{\Delta p^2} & \text{if the ahead object is near} \end{cases} \tag{25}$$

On a free road, the vehicle is asymptotically approaching its desired velocity. The corresponding acceleration is defined by Eq. (26), where a is the current acceleration, v is the current velocity, and v_c is the desired velocity.

$$\text{freeDriving} = a\left(1 - \left(\frac{v}{v_c}\right)^4\right) \tag{26}$$

In Algorithm 1, the functions $\sigma_a(x)$ and $\sigma_s(x)$ ensure that the computed acceleration and speed (given as parameters) are inside the ranges allowed by the physic description of the vehicle. After the *pool*'s vehicle has reached its goal, all the agents of the *pool* execute Activity 5.

3.5. Feedback activity

Feedbacks are computed at the end of the day according to the activities during the day, and are given to the carpoolers. This activity is the last of the day for a carpooling agent.

3.6. Environment updating activity

The environment updating is the activity that permits to a non-carpooling agent to register its mobility behavior in the environment in order to influence the driving simulation of the carpooler agents. Individual vehicles may be generated in the microscopic simulation model used in Activity 4. This activity is the last of the day for a non-carpooling agent.

4. Experimental results and implementation discussion

Experimentations were done on a population of 1000 people. One of the major goals of our experimentation is to compute and possibly optimize the solution time required to compute the (complex) agent-based interactions between people (nodes). One reason for doing this is to be able to replay reality and accurately predict carpooling negotiation outcome in order to dispose of a sufficient synthetic population to exercise a global carpooling matching advisor software. The trips are extracted from the data into the FEATHERS framework. The considered region is Flanders, Belgium. Fig. 9 illustrates the simulator windows: the lower-left window shows the roads, and the upper-left window displays the parameters of the simulation.

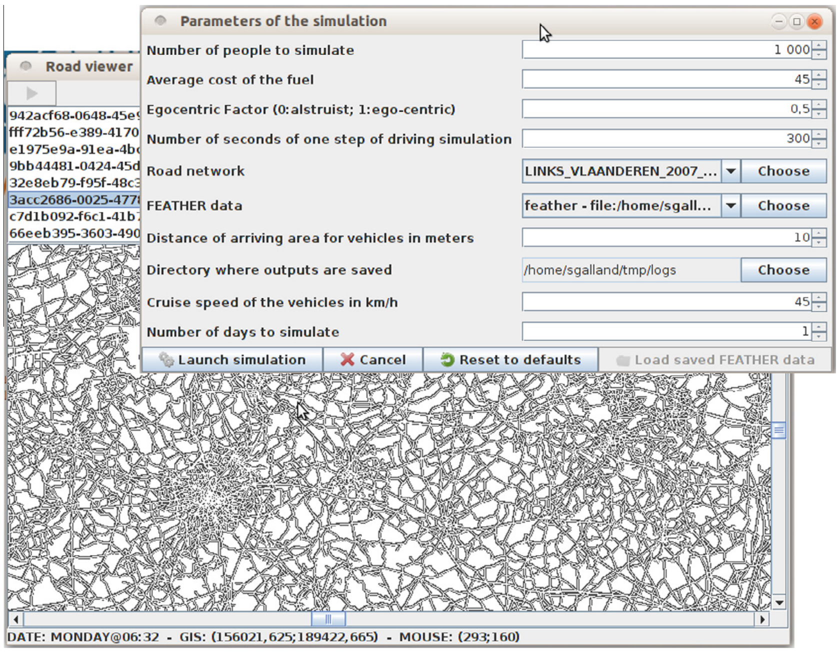

Fig. 9. Screenshot of the simulator.

The first experimentation is the proportion of drivers and passengers for every quarter of hours during the reference day. Fig. 10 gives the histograms for each category. 2/3 of the population are driving; 1/3 of the population is composed of passengers. The results obtained using our agent-based simulations are very interesting as they are close to reality, which can be compared with the original data extracted using the FEATHERS framework. Indeed, during the negotiation process, the pool partners do not change the time window of their trips.

Fig. 11 gives the histograms for the distribution among the transport modes for every quarter of hours during the reference day. 65% of the people are traveling in single vehicles; 31% are using the public transports; and 4% are carpooling. The average number of people per carpooling pool is of 2 with standard deviation of 1. The traffic density is still high during the high-activity periods: during the negotiation, the partners are not accepting enough changes in the time window of their trips. Therefore, they prefer to select the single-vehicle or the public transport modes than carpooling.

Fig. 12 gives the average computational times for the simulation of a day on an Intel Core i7 CPU 960 at 3.20 GHz, four cores, with Windows Seven (64 bits). Because the JaSim environment model is based upon the Influence-Reaction model (Michel, 2007), which permits to handle and solve the conflicts among simultaneous actions, the scheduling of the agents is a standard loop (each loop represents two seconds during the day). This approach does not cause a causality problem among the agent actions during the simulation. For the first curve, the people who do not want to carpool are also simulated during the trip execution. For the second curve, only the carpoolers are simulated.

Our simulator may be improved on several points: (i) the quality of the simulation's results, and (ii) the global performance of the simulator. The algorithms deployed in our model are designed according to the simplicity principle, and are easily replaceable by any other (compatible) negotiation protocols available to the agent-based community. The global

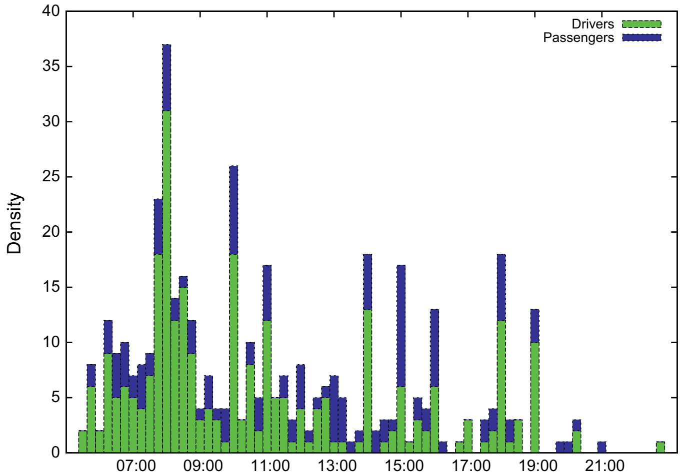

Fig. 10. Traffic density for each quarter of hours during the reference day, and per people category.

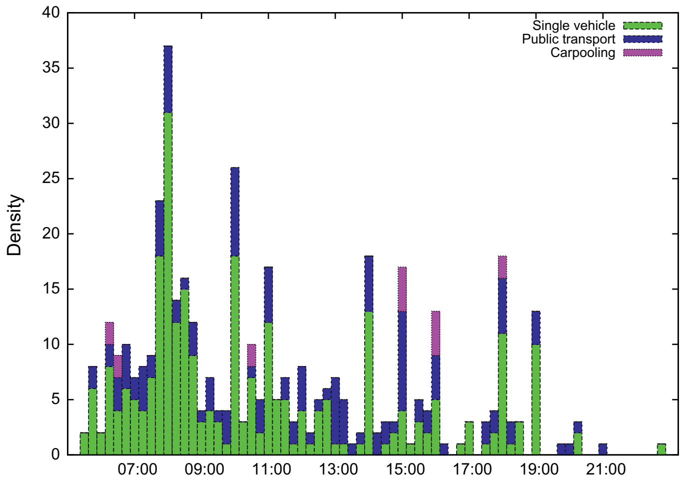

Fig. 11. Traffic density for each quarter of hours during the reference day, and per transport mode.

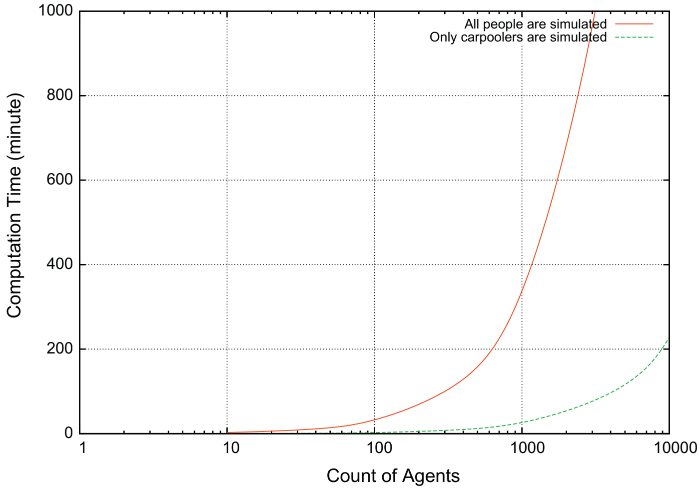

Fig. 12. Average computation time for a day.

exploration is skipped in this study because we do not have an online system for carpooling in place currently. We have presented the details of our approach for global exploration earlier in our paper and we intend to work on it as a part of the future work. The microscopic simulation of the vehicles may be replaced by a mesoscopic or macroscopic simulation model to improve the performances of this module. Performance is a crucial issue because the aim of the agent model is to serve as an active synthetic population to exercise the global advisor software. Wall clock time available for testing is limited and large agent communities are required for realistic evaluation. Hence, performance issues will get priority focus. In the case of a microscopic simulation, the agent scheduler used on the Janus platform may be replaced by an asynchronous scheduler (based on operating-system threads). This solution enables to run groups of agents in parallel in place of the current sequential approach. The current implementation is a proof-of-concept. All the algorithms must be tuned and redesigned to improve their performances. Our model and its implementation reproduces the behavior of a population in a carpooling problem. The microscopic simulation permits to reproduce the trip execution with high level of details (e.g. traffic jams).

5. Conclusion

In comparison to the state-of-the-art research work, the work/study presented within this paper has a number of advantages. Some of them are (i) the ability to analyze various effects of agent interaction with a detailed view on both communication and negotiation aspects, and (ii) the ability to simulate learning, adaptation and behavior reproduction of agents through modeling their interactions. Our simulation model on the Janus platform provides a solution to a complex simulation but needs a lot of computing resources (e.g. processing time, memory, and data storage) because of the high number of agents to simulate, and the big data processing for each agent. The agent-based model described constitutes an active synthetic population that can be used to exercise a *global matching advisor for carpooling*. Such advisor consists of complex mechanisms, including machine learning. Furthermore, it requires a critical mass of customers to operate efficiently. Exercising the advisor is useful for testing and for transient phenomena analysis. Like all models that cover real-world transportation problems, the simulation model proposed in the paper: (i) requires a large amount of detailed and accurate input data, including agent's socio-economic attributes and road network information and (ii) has scalability issues that are still to be solved. Indeed, it is necessary to consider a sufficiently large region to evaluate the carpooling process. Apart from scalability issues, future research will focus on enhancing the procedure for negotiation between agents.

Acknowledgments

Part of this work carried out under the European FP7 Open-FET project Data Science for Simulating the Era of Electric Vehicles (DATASIM) (project number FP7-ICT-270833).

This material is partly based upon the library JaSim, supported by Voxelia SAS. The views and conclusions contained within this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Voxelia SAS.

References

Bellemans, T., Kochan, B., Janssens, D., Wets, G., Arentze, T., Timmermans, H., 2010. Implementation framework and development trajectory of feathers activity-based simulation platform. Transport. Res. Rec.: J. Transport. Res. Board, 111–119.

Bellemans, T., Bothe, S., Cho, S., Giannotti, F., Janssens, D., Knapen, L., koerner, C., May, M., Nanni, M., Pedreschi, D., Stange, H., Trasarti, R., Yasar, A., Wets, G., 2012. An agent-based model to evaluate carpooling at large manufacturing plants. In: Procedia Computer Science, Procedia Computer Science. Elsevier, Niagara Falls.

Bernhardt, Kristen L., 2007. Agent-based modeling in transportation. Transport. Res. E – Circ., 72–80.

Cho, S., Yasar, A.U.H., Knapen, L., Bellemans, T., Janssens, D., Wets, G., 2012. A conceptual design of an agent-based interaction model for the carpooling application. In: 1st International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications. Niagara Falls, Canada.

Cossentino, M., Gaud, N., Hilaire, V., Galland, S., Koukam, A., 2010. ASPECS: an agent-oriented software process for engineering complex systems – how to design agent societies under a holonic perspective. Auton. Agents Multi-Agent Syst. 2, 260–304. http://dx.doi.org/10.1007/s10458-009-9099-4.

Dechter, R., Pearl, J., 1985. Generalized best-first search strategies and the optimality of A*. J. ACM 32, 505–536, http://doi.acm.org/10.1145/3828.3830, doi:http://dx.doi.org/10.1145/3828.3830.

Delling, D., Sanders, P., Schultes, D., Wagner, D., 2009. Engineering route planning algorithms. In: Lerner, J., Wagner, D., Zweig, K. (Eds.), Algorithmics of Large and Complex Networks, Lecture Notes in Computer Science, vol. 5515. Springer, Berlin Heidelberg, pp. 117–139. http://dx.doi.org/10.1007/978-3-642-02094-0_7, http://dx.doi.org/10.1007/978-3-642-02094-0_7.

DeLoach, S.B., Tiemann, T.K., 2012. Not driving alone? American commuting in the twenty-first century. Transportation 39, 521–537. http://dx.doi.org/10.1007/s11116-011-9374-5, http://dx.doi.org/10.1007/s11116-011-9374-5.

Ferber, J., 1999. Multiagent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley Professional.

Ferber, J., Michel, F., Baez, J., 2006. AGRE: integrating environments with organizations. In: Weyns, D., Parunak, H.V.D., Michel, F. (Eds.), Third International Workshop (E4MAS 2006), Lecture Notes in Artificial Intelligence, vol. 4389. Springer, Hakodate, Japan, pp. 48–56.

Galland, S., Gaud, N., Demange, J., Koukam, A., 2009. Environment model for multiagent-based simulation of 3D urban systems. In: the 7th European Workshop on Multiagent Systems (EUMAS09). Ayia Napa, Cyprus, Paper 36.

Galland, S., Gaud, N., Rodriguez, S., Hilaire, V., 2010. Janus: another yet general-purpose multiagent platform. In: the 7th Agent-Oriented Software Engineering Technical Forum (TFGAOSE-10), Agent Technical Fora. Agent Technical Fora, Paris, France.

Galland, S., Gaud, N., Yasar, A.u.h., Knapen, L., Janssens, D., Lamotte, O., 2013. Simulation model of carpooling with the janus multiagent platform. In: Yasar, A.U.H., Knapen, L. (Eds.), 2nd International Workshop on Agent-based Mobility, Traffic, and Transportation Models Methodologies and Applications (ABMTRANS13). Elsevier, Halifax, Nova Scotia, Canada.

Gaud, N., Galland, S., Hilaire, V., Koukam, A., 2008. An organizational platform for holonic and multiagent systems. In: the Sixth International Workshop on Programming Multi-Agent Systems (ProMAS08), of the Seventh International Conference on Autonomous agents and Multiagent Systems (AAMAS). Estoril, Portugal, pp. 111–126.

Horvitz, E., Apacible, J., Sarin, R., Liao, L., 2005. Prediction, expectation, and surprise: methods, designs, and study of a deployed traffic forecasting service. In: UAI. AUAI Press, pp. 275–283, http://dblp.uni-trier.de/db/conf/uai/uai2005.html#HorvitzASL05.

Kamar, E., Horvitz, E., 2009. Collaboration and shared plans in the open world: studies of ridesharing. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 187–194, http://dl.acm.org/citation.cfm?id=1661445.1661476.

Kass, G.V., 1980. An exploratory technique for investigating large quantities of categorical data. Appl. Stat. 29, 119–127.

Koenig, S., Likhachev, M., 2005. Fast replanning for navigation in unknown terrain. IEEE Trans. Robot. 21, 354–363. http://dx.doi.org/10.1109/TRO.2004.838026.

Macal, C.M., North, M.J., 2005. Tutorial on agent-based modeling and simulation. In: Proceedings of the 37th Conference on Winter Simulation, Winter Simulation Conference, pp. 2–15 <http://dl.acm.org/citation.cfm?id=1162708.1162712>.

Massaro, D.W., Chaney, B., Bigler, S., Lancaster, J., Iyer, S., Gawade, M., Eccleston, M., Gurrola, E., Lopez, A., 2009. Carpoolnow – just-in-time carpooling without elaborate preplanning. In: Filipe, J., Cordeiro, J. (Eds.), WEBIST. INSTICC Press, pp. 219–224, <http://dblp.uni-trier.de/db/conf/webist/webist2009.html#MassaroCBLIGEGL09>.

Michel, F., 2007. The IRM4S model: the influence/reaction principle for multiagent based simulation. In: AAMAS '07: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. ACM, New York, NY, USA, pp. 1–3.

Niazi, M., Hussain, A., 2011. Agent-based computing from multi-agent systems to agent-based models: a visual survey. Scientometrics 89, 479–499. http://dx.doi.org/10.1007/s11192-011-0468-9, http://dx.doi.org/10.1007/s11192-011-0468-9.

Odell, J., Parunak, H., Fleisher, M., Brueckner, S., 2002. Modeling agents and their environment. In: Giunchiglia, F., Odell, J., Weiss, G. (Eds.), Agent-Oriented Software Engineering III. Springer-Verlag, NY, USA.

Smith, L., Beckman, R., Anson, D., Nagel, K., Williams, M., 1995. TRANSIMS: Transportation analysis and simulation system. In: Conference: 5. National Transportation Planning Methods Applications Conference, 17–21 April. Seattle, WA, United States.

Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E 62, 1805–1824. http://dx.doi.org/10.1103/PhysRevE.62.1805, http://link.aps.org/doi/10.1103/PhysRevE.62.1805.

Waraich, R., Charypar, D., Balmer, M., Axhausen, K., Waraich, R., Waraich, R., Axhausen, K., Axhausen, K., 2009. Performance improvements for large scale traffic simulation in MATSim. [Arbeitsberichte Verkehrs- und Raumplanung], ETH, Eidgenössische Technische Hochschule Zürich, IVT, Institut für Verkehrsplanung und Transportsysteme <http://books.google.be/books?id=Wj-MXwAACAAJ>.