A Generic Software Framework for Carsharing Modelling Based on a Large-Scale Multi-agent Traffic Simulation Platform

Mohamed Haitam Laarabi(✉) and Raffaele Bruno

Institute for Informatics and Telematics (IIT),
Italian National Research Council (CNR), Pisa, Italy
{haitam.laarabi,raffaele.bruno}@iit.cnr.it
http://www.iit.cnr.it/

Abstract. Over the last decade, numerous carsharing systems have been deployed around the world. Yet, despite this success, net profit margins of carsharing services are still insufficient due to a complicated demand modelling and high expenses for fleet redistribution. To address these problems, different carsharing paradigms (e.g., one-way versus free floating), operational models and pricing schemes have been proposed. In order to assess the effectiveness of these models and strategies, realistic simulation tools are needed that account for the main parameters that affect system performance. To this end, we have developed a generic software framework that caters for several flavours of carsharing services, such as hybrid systems where both one-way and free floating modes coexist. In addition, the proposed framework accounts for electric vehicles, power sharing capabilities, smart charging policies, booking services, fleet redistribution and membership management. Our tool is based on MATSim, an open-source platform for multi-agent traffic simulation. To validate our simulation model we will use a case study based on data from the 2006 Lyon conurbation household travel survey, which provides information about more than three million trips.

Keywords: Carsharing · Electric vehicle · Multi-agent systems · Traffic simulation · MATSim

1 Introduction

Worldwide, the sharing economy is rapidly gaining momentum and it is typically identified with an economic model in which communities of people share access to goods and services, beyond one-to-one or singular ownership. Sharing economy can take a variety of forms, but shared transport systems are one of the fastest growing trends in terms of users and revenues. A recent study by Roland Berger Strategy Consultants [3] has shown that the most popular shared transport services, namely carsharing, ride-sharing, bike-sharing and shared-parking, experience market annual growth rates between 20% and 35%; revenues are expected to reach between 2 and 6 billion dollars for 2020.

M.-R. Namazi-Rad et al. (Eds.): ABMUS 2016, LNAI 10051, pp. 88–111, 2017.
DOI: 10.1007/978-3-319-51957-9_6

Carsharing is not a novel concept but over the last decade, worldwide participation to carsharing has steadily grown and today carsharing services operate in hundreds of cities around the world. Various types of carsharing paradigms have been proposed, including two-way, one-way, and free floating systems. Both in two-way and one-way carsharing, shared vehicles can be picked up only at designated locations (called stations). In two-way carsharing (e.g., Zipcar, Modo), customers are required to drop-off the vehicle at the station where they have initially picked it up. This constraint is dropped in one-way carsharing. Examples of one-way carsharing are Autolib, Ha:Mo ride, CITIZ. A further step towards maximum flexibility for the users is represented by free floating carsharing (such as Car2go, DriveNow, Enjoy), in which the concept of stations disappears and cars can be picked up and dropped off in any public parking within the area covered by the service. While free floating systems provide customers with higher flexibility than station-based solutions, the latter have the advantage of facilitating access to parking spaces, which are typically scarce resources in congested and densely populated cities. It is worth mentioning that even free-floating carsharing systems have sometimes dedicated on-street and municipal parking-lot spaces within dense city regions.

Despite the success of carsharing services and the expected exponential growth of the carsharing market [3], the economic viability of carsharing services is still an open issue due to asymmetric demand-offer problems (i.e. the unbalanced offer and demand of vehicles) and the high costs for fleet redistribution. It is important to point out that the planning and operation of carsharing systems is a complex task because it is very difficult to accurately model the dynamics of demand and supply processes. In fact, the availability of vehicles in a carsharing system is intrinsically dependent on trips that are demanded by the customers and vice-versa. In addition, there are several operational conditions that add uncertainties to the system about the future location of vehicles, e.g., how different pricing schemes impact users' decisions. Several optimisation approaches have been proposed to decide how to relocate vehicles in order to satisfy future (known a priori or predicted) customer demands [10,13,18]. Dynamic pricing schemes have been also proposed to incentivise users to leave vehicles at locations in which there is a shortage of vehicles [5,15,16]. However, the efficiency of these solutions has not been convincingly demonstrated because existing analytical and simulation models of carsharing systems are limited in scope or based on simplified assumptions. Furthermore, they leave out some of the most important characteristics that affect the performance of real systems, such as service booking or traffic congestion.

The main purpose of this work is to develop a modular, expandable and easily customisable simulation model of carsharing systems, which considers all the main characteristics of existing and future carsharing services. Key features of our framework are summarised in Fig. 1 and described in the following. First, we base the implementation of our framework on MATSim [1], a popular open-source and agent-based traffic simulation platform, which supports dynamic traffic assignment, large scenarios and detailed modelling of transportation networks.

Our objective is to develop a relatively independent core model that includes all basic functionalities of a general carsharing system and to implement specific strategies as external components. In this way, the carsharing system can be easily customised without affecting the core model. It is important to point out that our simulation tool accounts for all the main characteristics of real carsharing services, including fleet redistribution and membership management, booking policies, and electric vehicles. Finally, our model is not limited to conventional station-based or free-floating carsharing systems, but it also includes emerging carsharing paradigms, such as hybrid systems, as well as new types of electric cars that can be stacked together. Specifically, in hybrid carsharing systems both free-floating and one-way usage modalities may coexist (e.g. if a station facility is full the customer may be allowed to leave the car on any available on-street parking). Furthermore, electric car prototypes have been recently released or are under development that can be folded together when parked or driven as a stacked train, such as MIT's Bit Car [17], EO Smart [4], or ESPRIT vehicle [8]. It is expected that the adoption of these vehicles in next-generation carsharing services can significantly improve the system manageability, for instance by allowing advanced power sharing policies or more efficient redistribution mechanisms. However, it is necessary to develop more advanced models to study the performance of such sophisticated carsharing systems, and this is one of the targets of our simulation platform.

As a preliminary validation of the capabilities of our contribution to the MATSim framework we used the city of Lyon as case study. Specifically, we set up a simulation scenario using data from the 2006 Lyon conurbation household travel survey, which provides information about more than three million trips, and data from the Bluely system, a full electric one-way carsharing service that is operated in the city of Lyon.

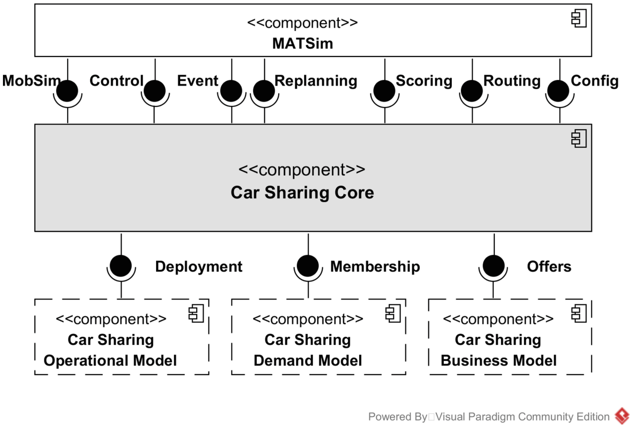


Fig. 1. An overview of carsharing system that consists of three main layers: MATSim, carsharing core and carsharing models

The rest of this paper is organised as follows. In Sect. 2 we overview the MATSim framework and discuss the limitations of existing models of carsharing. Section 3 is devoted to the description of the software architecture and functionalities of our proposed framework. Finally, Sect. 4 presents the case study we will be using for model validation and discusses the ongoing work.

2 Background on MATSim and the Carsharing Model

2.1 A Brief MATSim Description

The modelling of traffic simulation can be carried out at different levels of detail. One common approach is to model traffic as an aggregated flow of cars based on Origin-Destination matrix [14]. While it is a straightforward approach and does not require considerable computing resources, it does not allow the modelling of individuals' preferences and a detailed analysis of temporal-spatial traffic characteristics. In contrast, agent-based traffic simulation considers each individual as an agent, and in the case of MATSim the travel demand is described through an activity-based model [1]. This model describes individuals' travel choices with plans containing information on their daily activities such as time and location of activities to be performed and transport mode to be used in order to travel from one location to another. This activity-chain can be assigned to each individual with specific socio-demographic attributes. Then, simulation is executed to characterise the traffic interactions between the different individual travel choices, which are constrained by a space-time network. Each of these activity-chains or plans are evaluated with a score at the end of each iteration, which contributes to the selection of plans for the next iterations. The replanning concept is based on a genetic algorithm where only fittest plans are kept and might undergo mutations. The latter is a way for individuals to improve the score of their plans by varying, for instance, transport mode, routes or departure time. The simulation continues to iterate until it relaxes as depicted by Fig. 2. Finally, MATSim enables the simulation of large scale scenarios by leveraging queue-based models of traffic flows, which are significantly faster than microscopic continuous-time traffic models (e.g. car-following models) [12]. Specifically, each link of the road network is represented as a queue that adopts a First In First Out service discipline.

It is then possible, with such a traffic simulation platform, to account for specific attributes and mobility decisions that dynamically influence the travel choices of individuals, which is needed to accurately model a carsharing system. In the following, we will discuss the existing carsharing contribution in MATSim to clarify its base concept and features, as well as its limitations.

2.2 Current Carsharing Contribution

The carsharing model had been introduced into MATSim since 2008 [6] and it has been applied in different studies over the years, such as [5,7]. Three different types of carsharing services are considered by this existing carsharing contribution:

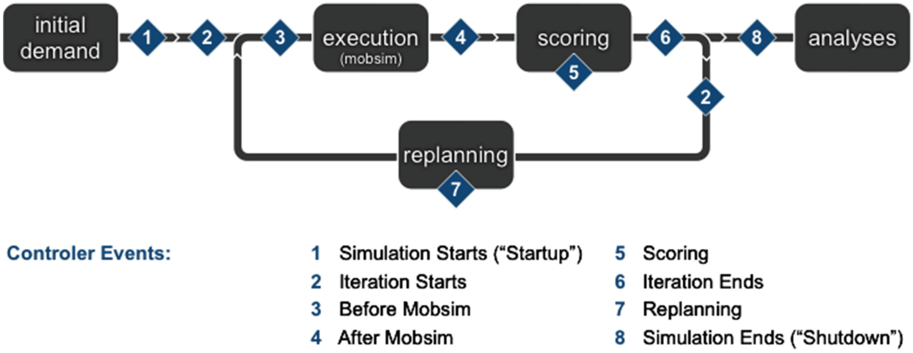


Fig. 2. The structure of the simulation controller of MATSim [19]

- Two-way, or round-trip, where a vehicle needs to be returned to the station from where it was picked up.
- One-way, where a vehicle can be dropped off at any station of the service.
- Free-Floating, where a vehicle can be picked up and dropped off at any station within the service area.

In the case of one-way carsharing service, the simulation model is portrayed graphically by Fig. 3 and summarised by the following steps:

1. *Booking Vehicle*: after the agent finishes an activity, it starts looking for the closest station, within a search distance radius, that has an available (i.e., non-booked) vehicle. If an available vehicle is found then the agent books it, and this makes the vehicle immediately unavailable to other agents;
2. *Access Walk*: agent walks from its current location (e.g. home) to the selected station;
3. *Pick Up*: agent picks up the vehicle and frees the parking spot;
4. *Booking Spot*: agent looks for the closest station to his final destination with an available parking space and books it, which makes the spot unavailable for others;
5. *Drive*: agent drives the vehicle to the destination station while interacting with other vehicles on the network;
6. *Drop Off*: Agent drops off the vehicle on the booked spot, which terminates the rental period;
7. *Egress Walk*: agent starts walking towards the location of his next activity;
8. Finally, agent carries out the remainder of the daily plan.

In case of either no vehicle is available or no parking spot can be found, the agent aborts its plan and consequently the controller assigns the worst score to the plan. The individual can also decide to use other modes such as public transport, bike or private car.

Regarding the behavioural model, agents use a scoring function that assesses their daily activity plans. In general, activities are evaluated positively with

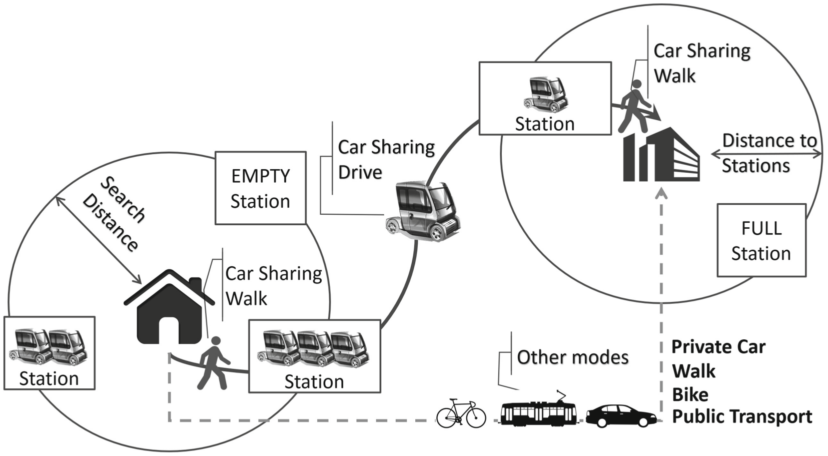


Fig. 3. Graphical representation of the simulation model of the carsharing contribution.

a utility function, whereas travelling is evaluated negatively with a dis-utility function [6]. As far as the carsharing transportation mode is concerned, the travel disutility function is composed of the following components:

- Access & Egress travel time cost;
- Carsharing usage constant;
- Rental time cost;
- Travel distance cost;
- Rental monetary cost;

In addition to the scoring, the behavioural model of the carsharing contribution is based on two different replanning strategies: *Carsharing Subtour Mode Choice Strategy* and *Random Trip To Carsharing Strategy*. The former one changes the transportation mode of all the legs of a sub-tour[1] to a different mode from a list of possible modes [6]. Note that certain transportation modes, called chain-based modes in MATSim, require that a specific resource (e.g., a private car or a bike) is available all along the sub-tour (e.g., an agent can not drive his car back from work to home if he had not previously parked it there). The second strategy is to incite individuals to use the carsharing service by substituting randomly a leg mode, which should not be a chain-based mode, by a carsharing mode.

[1] A sub-tour is any sequence of activities which starts and ends at the same location. For example, the chain home – work – shop – work – leisure – home (where both work activities are performed at the same location) contains two sub-tours: home-work-leisure-home and work-shop-work.

2.3 Limitations of Existing Carsharing Models

From a software engineering viewpoint, MATSim is a highly modular and customisable platform that solves dependencies problems by leveraging on dependency injection [20] using the Google Guice. The latter is a software framework that implements inversion of control for resolving dependencies [9]. It is important to point out that the modularity and extensibility of the simulation platform is critical for the carsharing modelling, because it allows a system designer to assess different operational, business or demand models without having to radically change the main components of the carsharing implementation. MATSim already offers interfaces for redefining the mobility simulation, events, scoring functions, routing modules and replanning strategies. The current carsharing contribution existing in MATSim extends the mobility simulation core to support the carsharing environment that was discussed above. However, there are no software hooks for providing customisable code at each step of the carsharing simulation, which would allow users to implement more easily different strategies for booking, pick up, drop off, access & egress and driving. Furthermore, carsharing stations are currently implemented as simple containers of vehicles. Whereas, stations represent a critical component of a carsharing system (e.g., to support smart charging policies of electric vehicles). Thus, it might be more appropriate to consider station as a special type of stage activity[2] within a complex trip. Finally, a realistic simulation of a carsharing system would require to consider also real-time fleet relocation, stations with charging infrastructure, more sophisticated booking and membership management mechanisms (e.g., for implementing dynamic pricing schemes).

Based on the limitations of the current carsharing contributions (both in terms of architecture and functionalities) we have designed a new simulation model of carsharing that ensures the independence of the operational, business and demand sub-models from the core components of the system (Fig. 1). In the following section, we describe in details the proposed framework.

3 A New Modelling Framework for Carsharing

3.1 Carsharing System Modelling

As explained above, the main goal of this work is to develop a simulation framework for carsharing that not only considers all main operational aspects of a real carsharing service, but should be also sufficiently flexible to accommodate the needs of next-generation carsharing systems (e.g. hybrid carsharing schemes, stackable vehicles, etc.). A prerequisite for flexibility is to design the corresponding software architecture in such a way that the “core” model is separated from the specific operational strategies. In fact, this approach facilitates system designers

[2] A stage activity is an activity part of a trip journey, such as public transport station, but is not considered as trip end as it is the case for default activities e.g. Home, Work etc.

to assess and compare different models. However, before discussing our proposed software architecture, let us first introduce all the key new features of our modelling framework for carsharing. Then, the following section will be devoted to explain how each of these general functionalities is implemented within the different software modules that compose our proposed carsharing model.

– Conventional, floating and charging stations: In our framework a station is a set of parking spaces that can be organised according to a specific physical layout and spatial constraints (e.g., to model a FIFO approach for vehicle pick up and drop-off). Each station keeps track of both demand and usage patterns and maintains information about vehicle availability and status (e.g., non-operational, booked, charging, etc.). A special type of stations, called floating stations, have been also implemented. Specifically, a floating station is a kind of virtual station that is used to model a vehicle dropped off anywhere else than a conventional station (e.g. to model a system that allows customers to leave the rented vehicles on on-street parking if the destination station is full). This features helps in modelling a variety of hybrid carsharing services in which both one-way and free floating carsharing approaches are employed. Finally, a station can be equipped with a charging infrastructure to enable the modelling of electric vehicles. We have implemented a variety of charging spot models, including multiple outputs multiple cables charging (MOMC) spots. MOMC spots have multiple cables which can charge several electric vehicles simultaneously, enhance the utilisation of the charging infrastructure and reduce investment costs [11]. Note that if a vehicle runs out of battery during the mobility simulation, then the plan is aborted and the vehicle disappears from the simulation.
– Electric and stackable vehicle: A shared vehicle can be an electric or hybrid car with associated specific energy consumption models. In case of electric vehicles, we assume that a vehicle is available only if the state of charge is above a certain threshold. An important novelty of our modelling framework is to allow the simulation of *stackable* vehicles, i.e., vehicles that can be mechanically and electrically connected and can be driven as a road train. This new generation of electric vehicles is expected to have a significant impact on the performance of future carsharing systems, especially for more efficient fleet relocation. For instance, in the European Project ESPRIT [8] is currently ongoing the prototyping of a lightweight electric vehicle for short trips in urban areas that can be stacked together in a road train of up to eight vehicles, seven being towed (see Fig. 4 for an illustration of the ESPRIT prototype vehicle). This makes easier

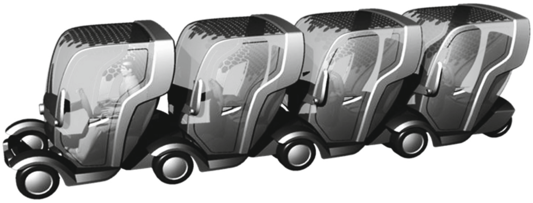


Fig. 4. ESPRIT stackable electric vehicles.

to redistribute them to locations where they are most wanted (a single staff can relocated multiple vehicles simultaneously). Furthermore, when parked at the stations they can be charged through the train electric connection and support dynamic load balancing. This can increase the carsharing operators' revenues by reducing the cost of installation (only one charging supply equipment to serve multiple parking spaces), while supporting, due to their small size, the growing demand for charging spots.

In practice, in our framework a stackable vehicle is modelled as a trailer that is assigned to a head vehicle to form a road train. Thus, when a road train is being relocated to a different station, the vehicle to be relocated should be detached from the station and attached to the head vehicle.

- Vehicle booking: The booking procedure is a critical management task for the carsharing provider. While the possibility to reserve a vehicle helps carsharing providers to predict future demands, they are also mandated to ensure vehicle availability at the requested time, which makes vehicle relocation even more compelling. In our framework, the booking procedures are executed during the mobility simulation for agents who have chosen to use carsharing as one of their transportation modes. We support two types of booking services: *early* booking and *immediate* booking. With early booking an agent can place a booking request a few hours before the desired starting time of the carsharing trip (e.g., up to 12 h). In this case, the booking process requires the starting time, as well as the source/destination of the trip. As better explained in the following, the booking system provides the customer with a single option or with multiple offers depending on the rental model and the relocation strategy that is implemented (user-based vs. operator based). It is worthwhile to mention that an available vehicle means that a vehicle is not booked, operational and its state-of-charge is above a critical threshold. Since both the source and destination are provided, the booking system can estimate the amount of required energy to successfully complete the trip. In case of insufficient battery, the system warns the agent during the booking process.

  With immediate booking, the agent searches for an available vehicle at the time he needs it. Then, he may decide to make a *Full Booking* with both a vehicle to pick up and a parking space at the destination. If the agent accepts one of the offers he received from the carsharing system, he should receive a confirmation and starts the access walk towards the source station. Otherwise, the user might ask for another offer. The agent can also decide to drive to his destination without pre-booking a parking space at the final station, or booking it later on during the rental period, which is considered as *Partial Booking*. At this stage the only required information is the source of the trip. In the case of immediate booking, the booked vehicles are immediately made unavailable for subsequent customers and considers the walking time needed to reach the start station. Note that the plan of the agent is aborted if he declines all received offers, unavailability of vehicles, booking time expiration, etc. The plan abortion results in assigning to that plan the worst score.
- Real-time vehicle relocation: The largest part of the carsharing modelling lies in the mobility simulation, where individuals have to book, search for available

vehicles, walk to/from stations and drive. In the mobility simulation is also implemented the modelling of the fleet relocation procedures. Two different approaches are modelled: operator-based and user-based [18]. In operator-based solution a separate staff is assumed that is dedicated to the relocation activities. In this case, the relocation strategy consists of decisions made by the system manager on which vehicles to relocate and how to assign staff to task relocation. The possibility to use stackable vehicles adds additional degrees of freedom (and complexity) in the decision process. On the contrary user-based relocation strategies make use of monetary incentives or bonus models for suggesting to customers alternative destinations than the preferred ones. It is also possible that an already driving customer is asked to pick up a second vehicle with him (taking advantage of the stackable capabilities of vehicles), thus also contributing to the rebalancing the system supply. To study the trade-off between incentive schemes for user relocation and staff planning for relocation is part of our future work.

- Rental model: The definition of different dynamic pricing schemes and rental models is supported to assess the impact of tariffs and monetary incentives discount on the performance of the carsharing service and its decision support system (e.g., user-based relocation, free-floating trips, multiple offers). In the rental model we also include the membership management. This includes the specification of the behavioural model of each customer, which specifies how that customer reacts to system offers and booking constraints.
- Demand model: Typically, the demand model is concerned with the generation/import of the transportation network, activity locations, synthetic population and their initial travel plans. We have extended the conventional demand model to include features that are relevant to the carsharing model, such as personal preferences of carsharing usage, characteristics of shared vehicles, station and charging infrastructure.
- Agent behaviour model: In our framework, the first stage of agents behaviour during mobility simulation is the *access walk*, which includes the walking leg towards the carsharing station. Before starting the walking leg, the agent performs the booking. Once an agent reaches the location of the selected vehicle, an event is triggered to start the second stage of the carsharing mobility simulation, i.e., the *carsharing drive*. As described above, in the case of immediate booking, the agent can provide a destination point but it is up to the agent to drop off the vehicle at the suggested stations or anywhere else. Furthermore, the carsharing system might invite an agent to take a second car with him. Then, the agent starts the driving leg towards the destination. The plan is aborted when an agent declines an offer, does not find a drop-off station or the vehicle runs out of energy. Once an agent drops off the vehicle the third and last stage of the carsharing mobility simulation starts, i.e., the *egress walk*. The agent starts walking to the next activity or next trip (in the case of last kilometre carsharing). The rental is ended and summarising trip information are logged.

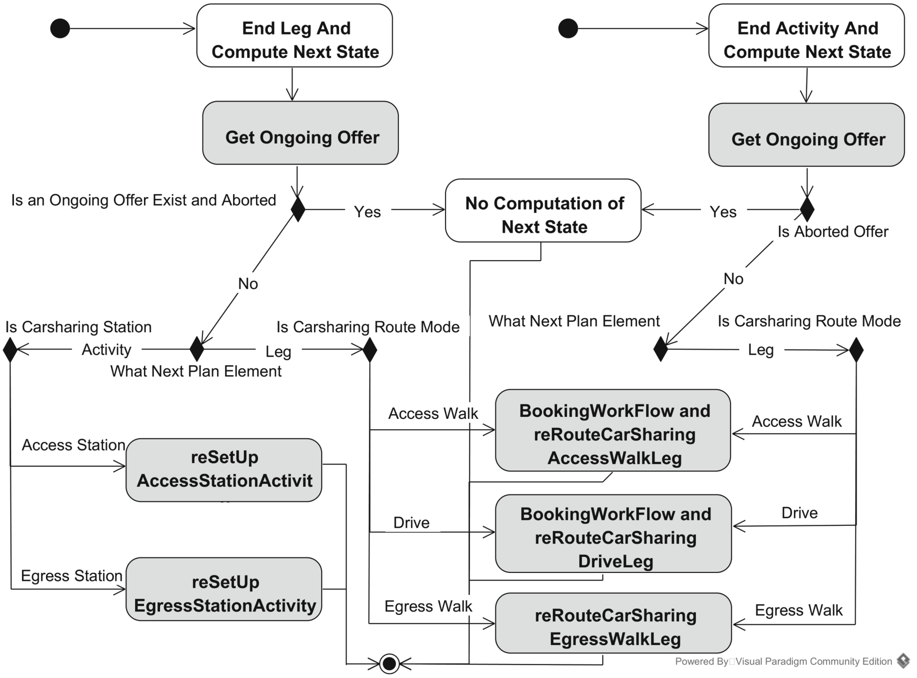

End Leg And Compute Next State
Get Ongoing Offer
Is an Ongoing Offer Exist and Aborted
Yes
No Computation of Next State
No
Is Carsharing Station
Activity
What Next Plan Element
Leg
Is Carsharing Route Mode
Access Station
reSetUp AccessStationActivit
Egress Station
reSetUp EgressStationActivity
Access Walk
Drive
Egress Walk
BookingWorkFlow and reRouteCarSharing AccessWalkLeg
BookingWorkFlow and reRouteCarSharing DriveLeg
reRouteCarSharing EgressWalkLeg
End Activity And Compute Next State
Get Ongoing Offer
Yes
Is Aborted Offer
No
What Next Plan Element
Is Carsharing Route Mode
Leg
Access Walk
Drive
Egress Walk
Powered By Visual Paradigm Community Edition

Fig. 5. General workflow of the agent's behaviour.

For the sake of completeness and to better illustrate the sequence of agent's decisions that are made during the mobility simulation, in Fig. 5 we show the workflow of the mobility simulation with a UML activity diagram. The white boxes refer to MATSim's default agent workflow, while the grey boxes refer to the carsharing sub-workflow. As shown in the diagram, each iteration of the mobility simulation consists in processing an element of the agent activity plan iteratively. Therefore, at the end of an activity or a leg the agent reflects about the next step. For instance, when an agent completes the execution of the access station activity, he has to pick up the car and starts driving. The agent behaviour model offers the flexibility not only to pick up and drop off in stations, but also nearby an activity location. Therefore, an agent can undertake not only direct trips — station to station — but also indirect trips where the agent can drop off vehicle nearby a shopping mall, for example, so that he picks it back up later on.

Conventional transportation modes (private cars, public transports, bikes, walking) are handled by the *MATSim default flow*. When a leg uses the carsharing mode the workflow described in Fig. 6 is invoked. First of all, the flow starts with booking verification since agents who choose plans that contain carsharing legs can decide to make an early booking or an immediate booking. In the former case, the simulation has already all the booking information (departure station, destination station, trip time), and the agent can immediately start the

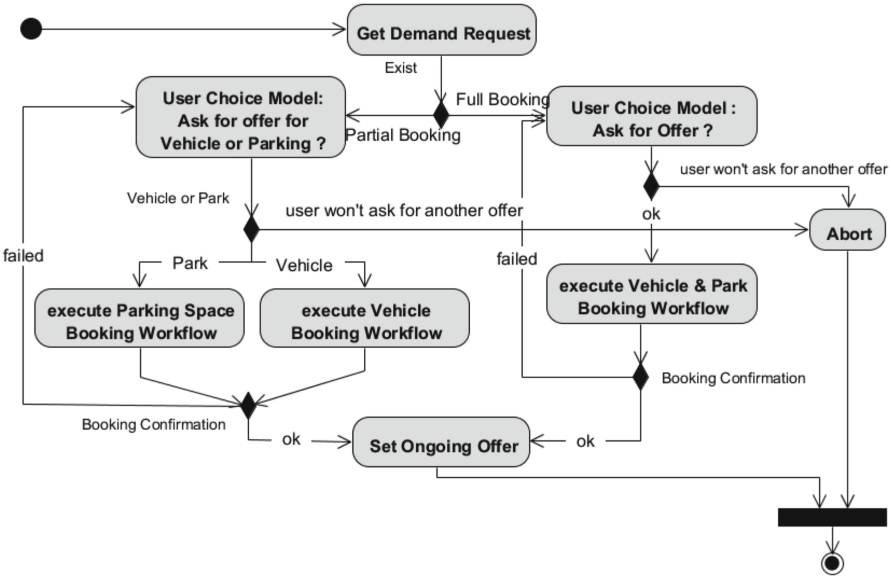

Get Demand Request
Exist
User Choice Model: Ask for offer for Vehicle or Parking ?
Full Booking
Partial Booking
User Choice Model : Ask for Offer ?
user won't ask for another offer
Vehicle or Park
user won't ask for another offer
ok
Abort
failed
Park
Vehicle
failed
execute Vehicle & Park Booking Workflow
execute Parking Space Booking Workflow
execute Vehicle Booking Workflow
Booking Confirmation
Booking Confirmation
ok
Set Ongoing Offer
ok

Fig. 6. Booking workflow.

access. In the latter case, which is shown in Fig. 6 the agent first searches for an available car, asks for an offer and books the vehicle if the offer is accepted. If the offer is declined or no car is available the travel plan is aborted and he obtains the worst score. Regarding the booking model, on the one hand the system can suggest personalised offers to each individual, for instance to incentivise users to participate to the vehicle relocation program. On the other hand, each agent is characterised by his own preferences regarding carsharing use. For instance, some agents would prefer to minimise travel costs by carsharing, which makes them more willing to drop-off the vehicle at a less favourite station. While other agents would prefer comfort and maintain their favourite departure and destination stations or accept to drive in a free floating mode.

3.2 Software Architecture

The design of a software architecture that is sufficiently flexible and independent from the carsharing operational model will require lightweight containers that enable to assemble components into a cohesive system. These containers are governed by a common pattern that characterises the way the components wiring is performed, and it is referred to as *Inversion of Control* (IoC) or more specifically as *Dependency Injection* [9]. This concept will be more apparent within the new modelling framework for carsharing with a UML Component Diagram, as shown in Fig. 7. Three main layers can be identified on the diagram. First, the MATSim layer is the base software component for traffic simulation, which, in turn, is based largely on the dependency injection design pattern [20]. The second

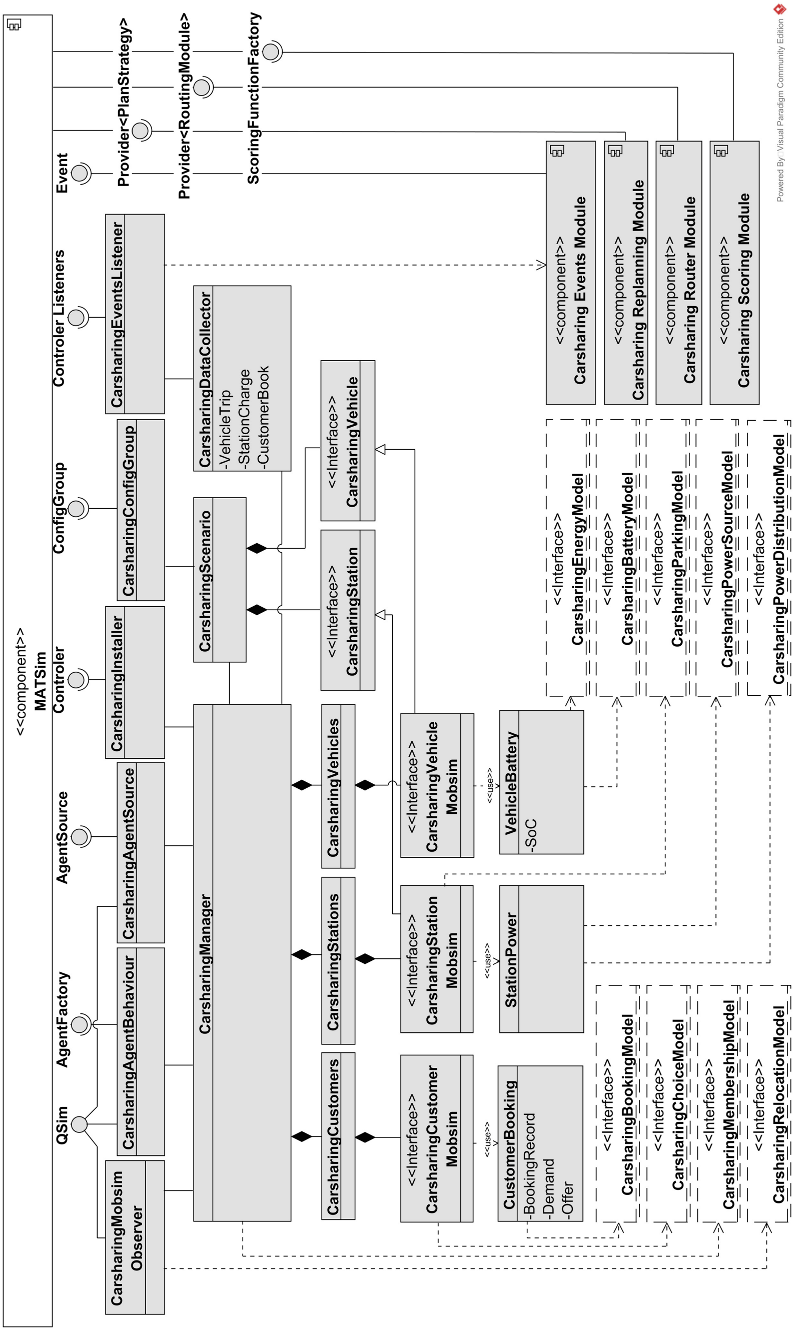

<<component>>
MATSim
Event
Provider<PlanStrategy>
Provider<RoutingModule>
ScoringFunctionFactory
Controler Listeners
ConfigGroup
Controler
AgentSource
AgentFactory
QSim
CarsharingEventsListener
CarsharingConfigGroup
CarsharingInstaller
CarsharingAgentSource
CarsharingAgentBehaviour
CarsharingMobsim Observer
CarsharingDataCollector
-VehicleTrip
-StationCharge
-CustomerBook
CarsharingScenario
CarsharingManager
<<Interface>> CarsharingVehicle
<<Interface>> CarsharingStation
CarsharingVehicles
CarsharingStations
CarsharingCustomers
<<Interface>> CarsharingVehicle Mobsim
<<Interface>> CarsharingStation Mobsim
<<Interface>> CarsharingCustomer Mobsim
<<use>>
VehicleBattery
-SoC
StationPower
CustomerBooking
-BookingRecord
-Demand
-Offer
<<component>> Carsharing Events Module
<<component>> Carsharing Replanning Module
<<component>> Carsharing Router Module
<<component>> Carsharing Scoring Module
<<Interface>> CarsharingEnergyModel
<<Interface>> CarsharingBatteryModel
<<Interface>> CarsharingParkingModel
<<Interface>> CarsharingPowerSourceModel
<<Interface>> CarsharingPowerDistributionModel
<<Interface>> CarsharingBookingModel
<<Interface>> CarsharingChoiceModel
<<Interface>> CarsharingMembershipModel
<<Interface>> CarsharingRelocationModel
Powered By: Visual Paradigm Community Edition

Fig. 7. UML class diagram of the carsharing system.

layer consists of the core modules (the grey boxes in Fig. 7) of the carsharing model, which set the environment for the simulation. In other words, it is an interface that translates the carsharing operational models into a traffic simulation, which is then controlled by the specific rules of the carsharing system. The third layer is a set of interfaces ready to be implemented (dashed boxes in Fig. 7) that describe the operational models the system designer wants to assess, the demand model to be simulated and the customer choice model to be considered. These three layers are wired by inversion of control, so that MATSim retains the control to the carsharing system and, in turn, it injects dependency into the third layer. In addition, every component of the system core is disaggregated and control is centralised within the carsharing manager. In the following we present the main software components of our framework.

- Carsharing Scenario: Represents the data structure that should be constructed, in principle, before installation of the carsharing module. It contains initial deployment of stations and vehicles, carsharing configuration module, car network and MATSim scenario.
- Carsharing Manager: It serves as an access point and for the carsharing scenario. Basically, it contains a mapping of customers and vehicles, in addition to a tree data structure for stations with geolocated information. Each of the three key features of the carsharing systems (customers, fleet, stations) are represented with a generic interface that injects dependency into the multiple instances that implement those interfaces. For instance, a station can be either a conventional charging station or a floating station but its representation is based on the same model. Stations are also characterised by power supply information and a vehicle container model.

  Similarly, vehicles can have different features too with a specific energy consumption and balancing model. Customers entity help to keep track of the carsharing usage, since the customer is identified by an id, geographic coordinates and socio-demographic information which can be monitored not only over iterations but even over entirely different simulations set-ups. The manager has also access to all the models, booking service as well as to the logging service that generates the carsharing event file for post-analysis.
- Carsharing Agent Behaviour: Describes an agent behaviour work flow, therefore it has access to the carsharing manager and governed by the mobility simulation.
- Carsharing Agent Source: Responsible for physical deployment of carsharing fleet in the car network.
- Carsharing Events Listener: It captures fired carsharing event, during mobility simulation. Data collection is basically undertaken at this level. This component can be easily extended through class inheritance.
- Models: They represent interfaces for energy, relocation, demand, rental, user choice and vehicle models, which help at injecting customized models into the carsharing core.
- Carsharing Mobsim Observer: This class offers plan view of the simulation and run in parallel with mobility simulation steps. It can be extended to included further controls and handling, as the operator-based relocation.

- Carsharing Installer: A utility class for installation of the carsharing module. It can be customized by inserting new models, observer, and/or events listener.
- Carsharing Data Collector: By default, this class collects information about trips, charging at stations and customer booking. Then it writes them out in a file predefined in the carsharing configuration group.
- Component modules: These components offer direct usage of MATSim's replanning, routing and scoring modules, and can be easily customized.

4 Validation and Ongoing Work

To validate our modelling approach of carsharing systems we use a real-world scenario for the city of Lyon. Our test case is built using real traffic demands for the metropolitan area of Lyon based on data from the 2006 Lyon conurbation household travel survey. More precisely, the traffic demand is originally provided in terms of two OD matrices, representing the travel modes and travel purposes between 148 different traffic zones in the metropolitan area of Lyon. As shown in Fig. 8, we focus our study only on Lyon downtown, because this is the service area of Bluely, a one-way station-based carsharing provider that operates in Lyon. The simulated area includes 56 zones (in the figure zone borders are shown with black polygons). According to the travel survey, during a typical working day, almost 3 million trips have an origin or destination within the considered 56 zones.

As discussed [2], generating plausible daily activity chains from OD matrices is a complex task and it generally requires the integration of census and

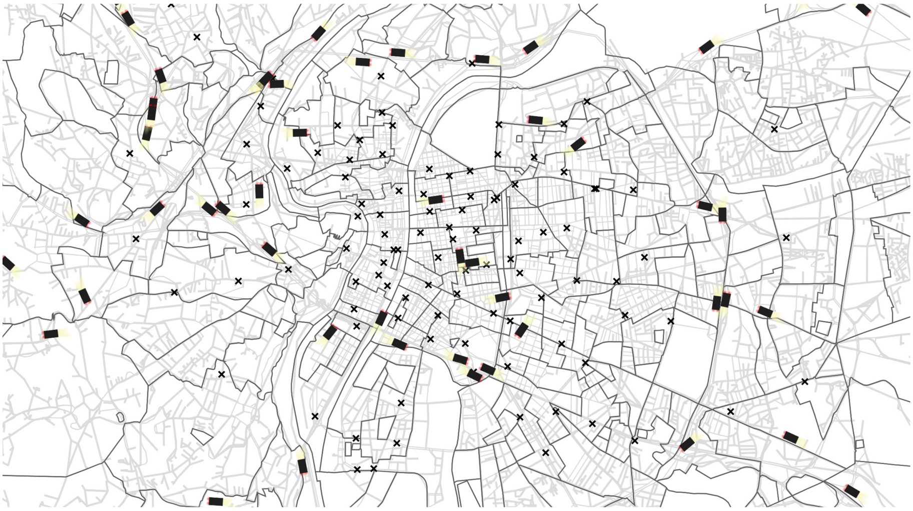


Fig. 8. Lyon Scenario: simulating the Bluely service in Lyon downtown. The diagram shows the road network in the Via traffic visualiser. Carsharing stations are depicted as crosses, while rectangles are shared vehicles.

socio-demographic data from various sources, as well as simplifying assumptions. For instance, census data can be used to generate a synthetic population, including population groups (e.g., children, workers, non-workers, pensioners), spatially distributed among zones and household compositions. Similarly business census data can provide the number of workplaces, as well as shopping, leisure and education facilities, which can be randomly deployed within each zone. For simplicity we have only used the OD matrices to derive a preliminary travel demand for the Lyon scenario. More precisely, OD matrices are used to derive the trip shares between each pair of zones and to assign a *purpose* to each trip (i.e., home-work-home, or home-education-home, and home-leisure-home). Then, we also assume that the temporal distribution of opening times and durations of each activity type is known. Finally, we assign an agent to each trip departing from a zone and we randomly locate the agent in the initial zone. The total number of agents in our simulation is 100,000. Following these steps, we have obtained the modal share depicted by Fig. 9.

Regarding the station infrastructure of the carsharing system, we consider the locations of the real Bluely stations (104 stations within Lyon downtown), all equipped with a charging point. In each scenario, the total number of parking spaces in the carsharing network is assumed to be two times the fleet size. Then, parking spaces are uniformly assigned to each station. The carsharing membership is assumed to be a function of the distance to the carsharing stations: all agents who execute their activities in facilities which have one or more stations within a predefined search distance, are considered to be members and potential users of the carsharing system. Specifically, the considered search distance in this case study is 500 m, which made 20% of the simulated agents holders of the

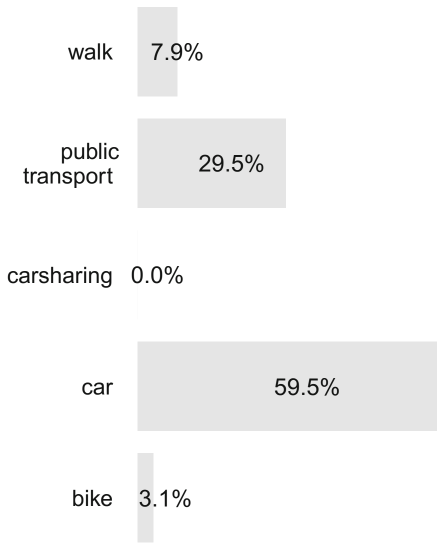
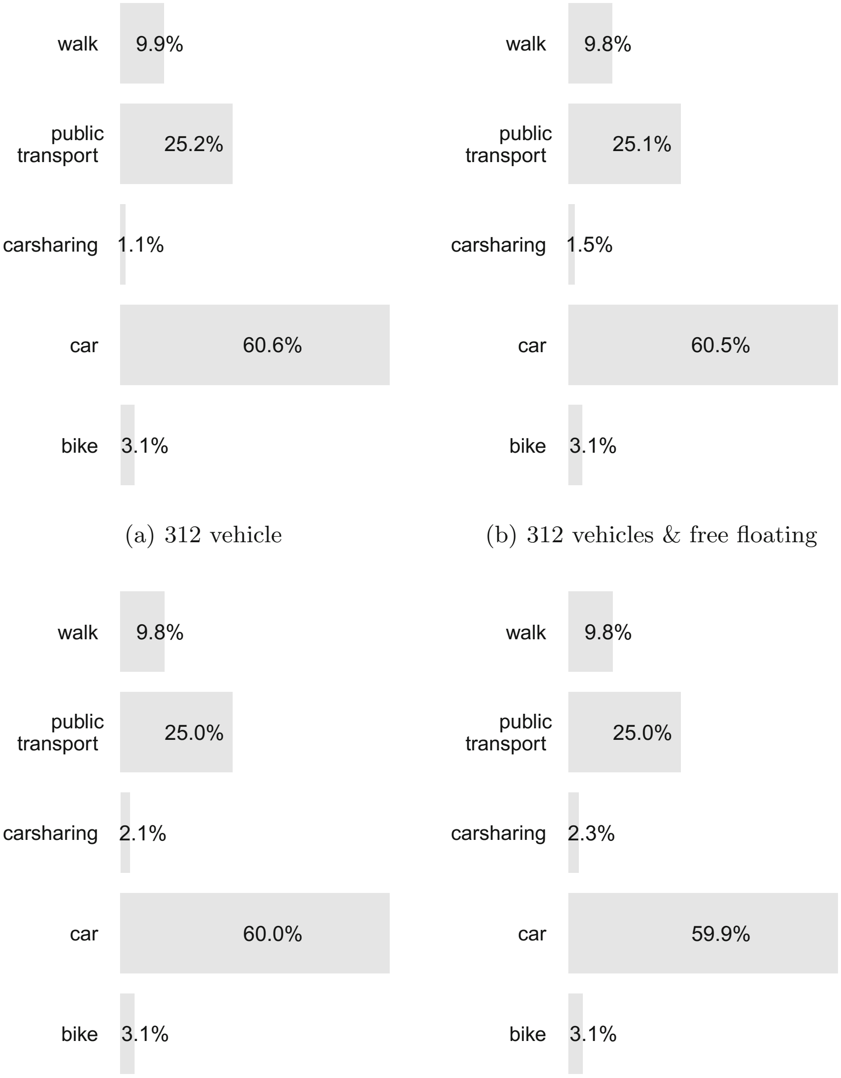


Fig. 9. Initial modal share



(a) 312 vehicle

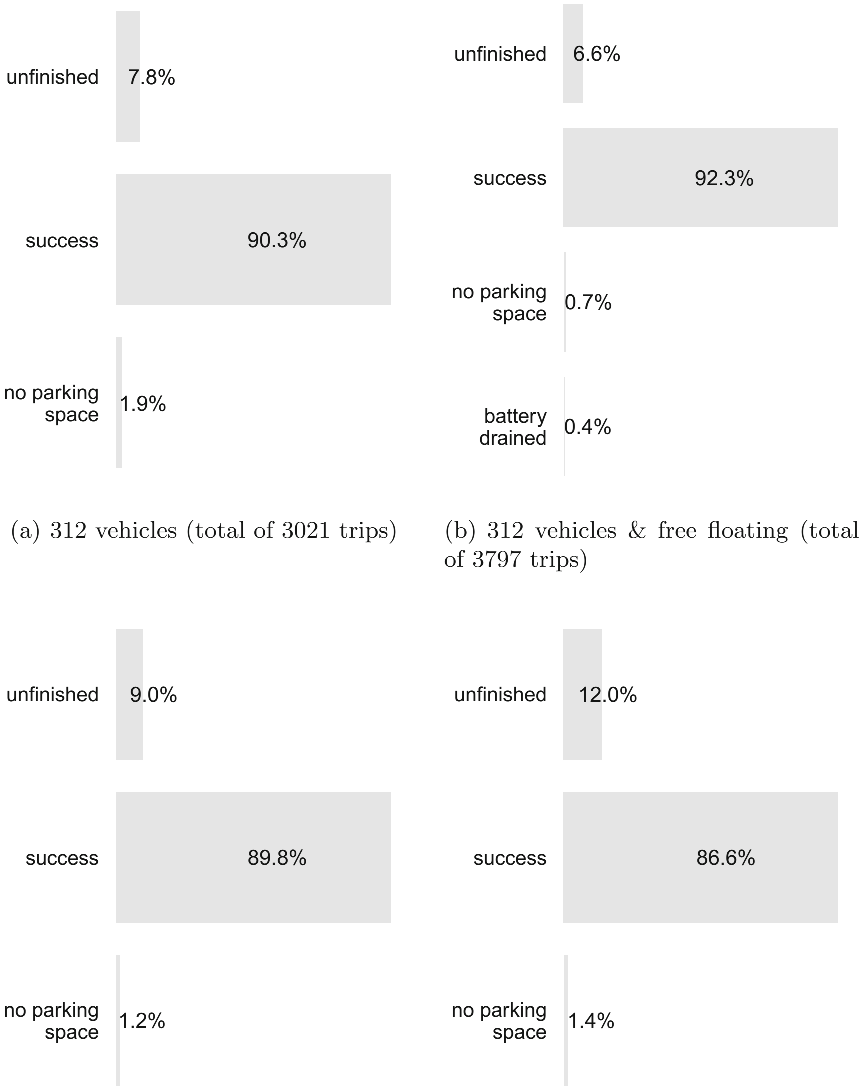
(b) 312 vehicles & free floating

(c) 624 vehicles

(d) 936 vehicles

Fig. 10. The modal share after inciting carsharing members to use the carsharing mode, through a random mode choice as a replanning strategy to explore various solutions and combinations. The subtitle of each sub figure refers to the fleet size as a unique input variable to compare the different scenarios.

carsharing membership. Finally, we compare three different scenarios in which three, six and nine vehicles per station are deployed, as well as a fourth scenario where free floating is also allowed. The objective is to observe the performance of the carsharing system for various system capacities.



(a) 312 vehicles (total of 3021 trips)

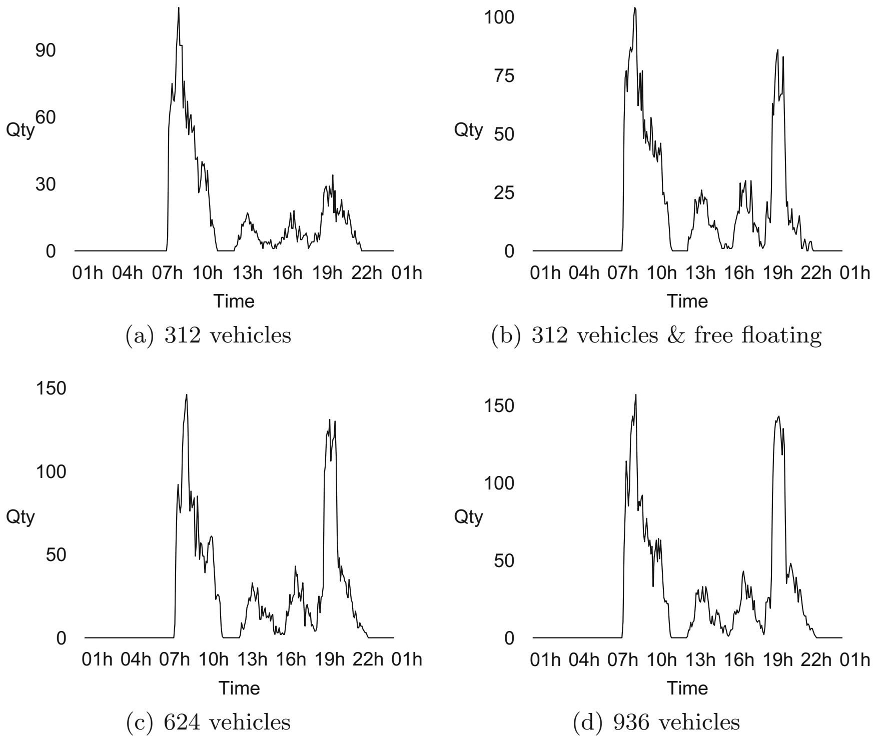
(b) 312 vehicles & free floating (total of 3797 trips)

(c) 624 vehicles (total of 5691 trips)

(d) 936 vehicles (total of 6394 trips)

Fig. 11. Four pie charts describing the relative number of successful trips to the ones that failed due to 2 - unavailability of parking space or 4 - battery drained or 3 - other reasons such as end of simulation horizon.

The free floating capability has been constrained in such a way to ensure maximum performance in the absence of operator/user-based relocation. An agent, therefore, is constrained to opt for free floating offer if:



(a) 312 vehicles

(b) 312 vehicles & free floating

(c) 624 vehicles

(d) 936 vehicles

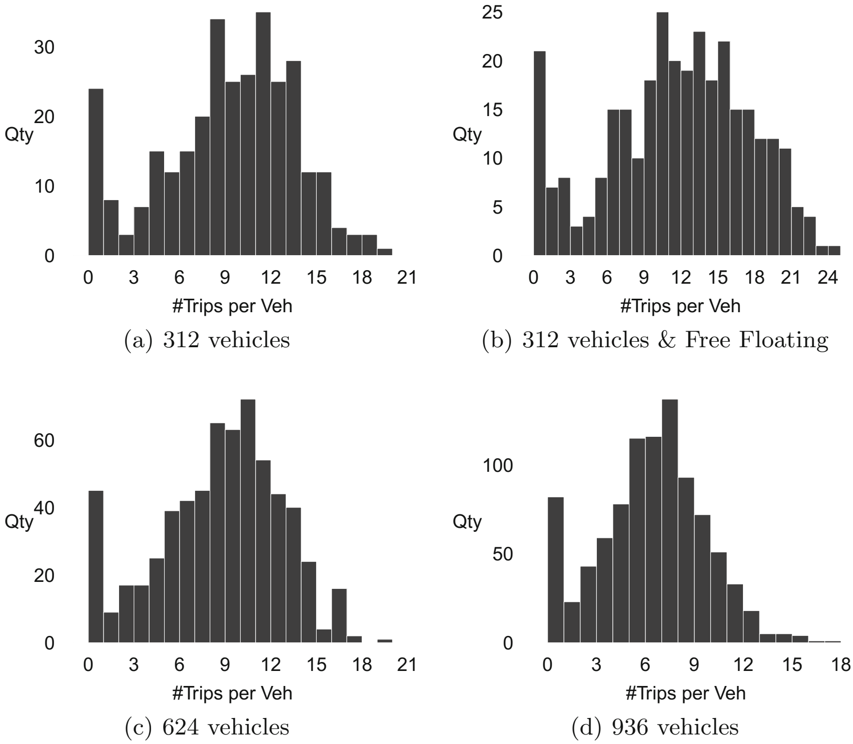
Fig. 12. Four time-based histograms, for each scenario, depicting the number of simultaneous trips (or En Route) throughout the simulation day, per a bin of time of 15 min.

– it does not find a parking station at the destination;
– it finds an on-street parked vehicle within a search distance range.

These constraints can be relaxed once relocation strategies are included in the operational model. Where the economic dimension might drive the decisions to optimize such a hybrid carsharing model.

The results obtained from the simulation of the above-described scenarios are reported in Figs. 10, 11, 12, 13 and 14.

First of all, we start by analysing the change in the modal share after introducing carsharing, which is reported by the Fig. 10. The results show that the carsharing mode is a *weak signal* in the transportation network, as the usage of carsharing is limited by the relatively small fleet of vehicles with respect to the number of travellers. We can also observe the proportional increase in modal share with respect to the increase of the carsharing fleet. For instance the modal share *doubles* when we doubled the fleet size to 624 vehicles, while it increased by 120% when fleet size was expanded to 936 vehicles. From operational perspective, one can observe a similar behaviour to a real world carsharing system,



(a) 312 vehicles

(b) 312 vehicles & Free Floating

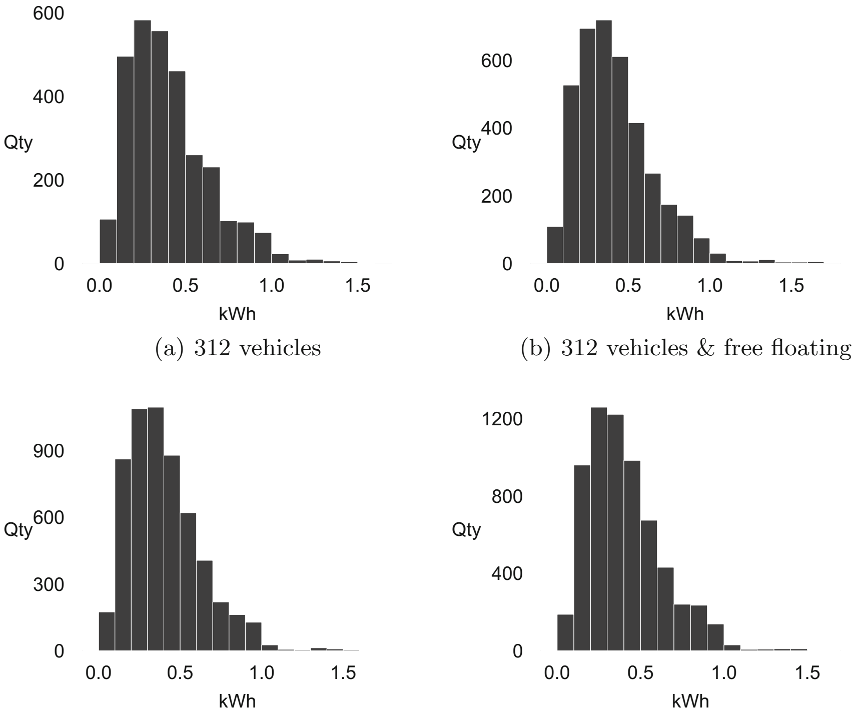
(c) 624 vehicles

(d) 936 vehicles

Fig. 13. Four histograms, for each scenario, depicting the number of trips (or rotation) per vehicle.

where it is not necessary to have a linear relationship between the size of the deployed carsharing fleet, and the carsharing usage/trips.

Regarding the scenario where agents are constrained to opt for dropping off or picking up an on-street vehicle, as detailed beforehand. The modal share also increased by 40% as shown on Fig. 10(b). This increase can be explained by the fact that free floating option has relaxed the constraints of necessity to drop a vehicle in one of the 104 carsharing stations. Thus, more agents were able to use carsharing system because they were allowed to drop the vehicles anywhere in the studied area. Since all the carsharing members have at least one station at proximity of each of the facilities where their activities are undertaken, the agents, then, were constrained to use a free floating vehicle because no parking space was available in the nearby station. In this way, even the on-street vehicles were dropped off within a radius of 500 m from a full station. Therefore, the agents willing to pick up a vehicle were again constrained to pick up from a station considered full as long as no on-street vehicle are parked in the neighbourhood.



(a) 312 vehicles

(b) 312 vehicles & free floating

(c) 624 vehicles

(d) 936 vehicles

Fig. 14. Four histograms, for each scenario, depicting the amount of energy expressed in kilowatt-hour, which was consumed during a single trip.

We support the modal share with further details on the carsharing trips that actually have taken place after the replanning phase, as reported in Fig. 11. In fact, since there are 20% potential carsharing users, the number of agents who did not receive an offer after asking for the carsharing service, was not included in these charts. Having said that, we observe the evidence of clear increase of number of carsharing trips, as well as, the efficiency of the free floating scenario where relatively more successful trips have been recorded due to relaxation of the parking space constraint.

The results of Fig. 12 show the increasing number of simultaneous trips when expanding the fleet size. However, the highest number of simultaneous trips represent only 35%, 23% and 17% of the fleet, respectively, for the scenarios with 312, 624 and 936 vehicles. This means that the vehicles are not well distributed in the area, and the disparity grows higher by expanding the fleet while preserving the same distribution. This is another observed fact in real world systems, where spatial as well as temporal distribution of vehicles are key assets for the optimisation of the operational model.

Giving the simulated scenarios, a decision maker would wish to use this simulator to evaluate and measure the success of certain relocation strategies, either using a staff or through encouraging users to choose specific stations or to take a second car as a road train. Successful relocation algorithms would lead to an increase of carsharing usage, and notably to the increase of simultaneous trips, as an indicator of demand absorption during the day.

The trips histograms in Fig. 13 confirm the analysis drawn from the previous results. We note that, the number of rotations per vehicle is relatively high when comparing to real world systems. In other words, same vehicles are being reused again and again, while some are never used. Furthermore, we observe that the number of rotations decreases when the fleet is larger. While free floating increases the usage and rotation per vehicles due to the aforementioned constraints.

At last, the histograms in Fig. 14 report the amount of energy consumed per trip. These results are essential for vehicle design when it comes to battery capacity and usage. It is interesting to note that there are negligible differences between the results for different fleet size. This can be explained by observing that the energy consumption mainly depends on the mobility patterns of the carsharing trips. Thus, increasing the fleet size has not a significant impact on the length or travel time of carsharing trips.

5 Conclusion

The main contribution of this work is the development of a new simulation model of carsharing for MATSim, an urban-scale and activity-based multi-agent modelling framework of multi-modal transportation systems. Our module supports the modelling of various interrelated components of generic carsharing systems, including booking services, fleet redistribution, charging policies and membership management. In addition, we also included new carsharing paradigms such as mixed systems (free-floating and station-based), and stackable vehicles. To evaluate our model we used a scenario with real travel data from the city of Lyon.

As an ongoing work we intend to use our modelling framework for providing an initial estimate of the usage patterns of the carsharing service under different operational parameters and procedures. Specifically, we will assess the impact of the charging infrastructure and vehicle parameters (i.e., number of charging points, maximum charging power, battery capacity) on the operational time of electric shared vehicles. Our findings can provide a guidance to the system designers for deriving an optimal configuration of the charging station to minimise investment costs. In addition, we will provide a first analysis of simple heuristics for vehicle relocation. A first intuitive approach would be to relocate vehicles by moving them from full stations to empty stations at certain times of the day (e.g., peak hours) if the driving distance is below a given threshold (e.g., to limit battery consumption and to guarantee fast relocation). Alternatively, relocation could be carried out by those users, whose destinations are close to an area/station with an insufficient supply of vehicles. The amount of monetary incentives that are needed to encourage such users to participate in the relocation activity could be easily estimated.

Acknowledgement. This work has been partially funded by the ESPRIT project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 653395.

References

1. Balmer, M., Cetin, N., Nagel, K., Raney, B.: Towards truly agent-based traffic and mobility simulations. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 60–67, AAMAS 2004. IEEE Computer Society, Washington, DC, USA (2004). http://dx.doi.org/10.1109/AAMAS.2004.285, 00098
2. Balmer, M., Rieser, M., Vogel, A., Axhausen, K.W., Nagel, K.: Generating day plans based on origin-destination matrices. In: 5th Swiss Transport Research Conference, March 2005
3. Berger, R.: Roland Berger study on the market for car sharing in China: major potential for vehicle manufacturers and service providers|Press room|Roland Berger, July 2014. http://www.rolandberger.com/press_releases/Car_Sharing_in_China_2014.html
4. Birnschein, T., Kirchner, F., Girault, B., Yuksel, M., Machowinski, J.: An innovative, comprehensive concept for energy efficient electric mobility - EO smart connecting car. In: 2012 IEEE International Energy Conference and Exhibition (ENERGYCON), pp. 1028–1033, September 2012. 00003
5. Ciari, F., Balac, M., Balmer, M.: Modelling the effect of different pricing schemes on free-floating carsharing travel demand: a test case for Zurich, Switzerland. Transportation **42**(3), 413–433 (2015). http://link.springer.com/10.1007/s11116-015-9608-z
6. Ciari, F., Balmer, M., Axhausen, K.W.: Concepts for a large scale car-sharing system: modeling and evaluation with an agent-based approach. In: 88th Annual Meeting of Transportation Research Board, January 2009. http://www.researchgate.net/profile/Kay_Axhausen/publication/228952222_Concepts_for_a_large_scale_car-sharing_system_Modelling_and_evaluation_with_an_agent-based_approach/links/0deec517bbebe35452000000.pdf
7. Ciari, F., Schuessler, N., Axhausen, K.W.: Estimation of carsharing demand using an activity-based microsimulation approach: model discussion and some results. Int. J. Sustain. Transp. **7**(1), 70–84 (2013)
8. ESPRIT Project 2016, E.U.: Easily distributed personal rapid transit. http://www.esprit-transport-system.eu/
9. Fowler, M.: Inversion of control containers and the dependency injection pattern, January 2004. http://martinfowler.com/articles/injection.html#ServiceLocatorVsDependencyInjection
10. Jorge, D., Correia, G.H.A., Barnhart, C.: Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. IEEE Trans. Intell. Transp. Syst. **15**(4), 1667–1675 (2014). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6754142

11. Lindgren, J., Lund, P.D.: Identifying bottlenecks in charging infrastructure of plug-in hybrid electric vehicles through agent-based traffic simulation. Int. J. Low-Carbon Technol. **10**(2), 110–118 (2015)
12. Meister, K., Balmer, M., Ciari, F., Horni, A., Rieser, M., Waraich, R.A., Axhausen, K.W.: Large-scale agent-based travel demand optimization applied to Switzerland, including mode choice. Technical report, Zürich (2010)
13. Nourinejad, M., Zhu, S., Bahrami, S., Roorda, M.J.: Vehicle relocation and staff rebalancing in one-way carsharing systems. Transp. Res. Part E: Logistics Transp. Rev. **81**, 98–113 (2015). http://linkinghub.elsevier.com/retrieve/pii/S1366554515001349
14. Ortuzar, J., Willumsen, L.G.: Modelling Transport, 4th edn. Wiley, New York (2011)
15. Salies, E.: Real-time pricing when some consumers resist in saving electricity. Energy Policy **59**, 843–849 (2013). http://linkinghub.elsevier.com/retrieve/pii/S0301421513003030
16. Soltani, N.Y., Kim, S.J., Giannakis, G.B.: Real-time load elasticity tracking and pricing for electric vehicle charging. IEEE Trans. Smart Grid **6**(3), 1303–1313 (2015). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6948246,00006
17. Vairani, F.: BitCar: design concept for a collapsible stackable city car. Thesis, Massachusetts Institute of Technology (2009). http://dspace.mit.edu/handle/1721.1/49717
18. Weikl, S., Bogenberger, K.: Relocation strategies and algorithms for free-floating car sharing systems. IEEE Intell. Transp. Syst. Mag. **5**(4), 100–111 (2013)
19. Zilske, M.: Controler structure. http://archive.matsim.org/book/export/html/271
20. Zilske, M., Nagel, K.: Software architecture for a transparent and versatile traffic simulation. In: Agent Based Modelling of Urban Systems, May 2016