

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

An Integrated Simulation Framework to Model Electric Vehicles Operations and Services

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

An Integrated Simulation Framework to Model Electric Vehicles Operations and Services / Bedogni, Luca; Bononi, Luciano; Di Felice, Marco; D'Elia, Alfredo; Mock, Randolph; Morandi, Francesco; Rondelli, Simone; Salmon Cinotti, Tullio; Vergari, Fabio. - In: IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. - ISSN 0018-9545. - STAMPA. - 65:8(2016), pp. 7152984.5900-7152984.5917. [10.1109/TVT.2015.2453125]

*Availability:*

This version is available at: <https://hdl.handle.net/11585/568953> since: 2016-11-22

*Published:*

DOI: <http://doi.org/10.1109/TVT.2015.2453125>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

**Bedogni, L., et al. "An Integrated Simulation Framework to Model Electric Vehicle Operations and Services." *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, 2016, pp. 5900-5917.**

The final published version is available online at :  
<http://dx.doi.org/10.1109/TVT.2015.2453125>

#### Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

*This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)*

***When citing, please refer to the published version.***

# An Integrated Simulation Framework to Model Electric Vehicles Operations and Services

L. Bedogni, L. Bononi, M. Di Felice, A. D'Elia, R. Mock, F. Morandi, S. Rondelli, T. Salmon Cinotti, F. Vergari<sup>1</sup>

**Abstract**— At present, battery charging operations constitute one of the most critical obstacles toward a large-scale uptake of Electric Mobility (EM), due to performance issues and implementation complexities. Although several solutions based on the utilization of Information and Communication Technologies (ICTs) and on mobile applications have been investigated to assist the Electric Vehicles (EVs) drivers and to coordinate the charging operations, there is still the problem of how to evaluate and validate such solutions on realistic scenarios, due to the lack of accurate simulators integrating vehicular mobility, wireless communication and battery charging/discharging models. In this paper, we attempt to fill this gap, by proposing a novel EV simulation platform that can assist the pre-deployment of charging infrastructures and services on realistic, large-scale EM scenarios. The simulation platform, realized within the ARTEMIS EU project “Internet of Energy for Electric Mobility” (IoE), supports two utilization modes, i.e. evaluation of EM scenarios and immersive emulation of EM-related mobile applications, thanks to a semantic architecture through which virtual and real components can be integrated in a seamless way. We provide three major contributions with respect to the state of the art. First, we extend the existing co-simulation platform composed by SUMO (vehicular traffic simulator) and OMNET++ (network simulator) with realistic models of EVs, EVSEs and ontology-based communication protocols that enable the deployment of city-wide mobile services (e.g. charging reservation). Second, we validate the battery model against the consumptions data of target EVs, and we evaluate the operations of EVs on a large-scale scenario (the city of Bologna), by analyzing the effectiveness of the charging reservation process and the resulting impact to the smart grid. Finally, we introduce the Mobile Application Zoo (MAZ), a sandbox through which EM-related mobile applications can be seamlessly integrated within the simulation platform in order to be validated on virtual environments before their deployment on real scenarios, and we describe the implementation of an Android app for battery monitoring and charging reservation.

**Index Terms**— Internet of Energy, Electric Vehicles, Modeling and Simulation, Mobile applications

<sup>1</sup> Sumbitted on June 2014.

L. Bedogni, L. Bononi, M. Di Felice and S. Rondelli are with the Department of Computer Science and Engineering, University of Bologna, Italy (e-mails: {lbedogni, bononi, difelice}@cs.unibo.it, simone.rondelli2@studio.unibo.it).

A. D'Elia, F. Morandi, T. Salmon Cinotti and F. Vergari are with ARCES, University of Bologna, Italy (e-mails: {adelia, fmorandi, tsalmon, fvergari}@arces.unibo.it).

R. Mock is with SIEMENS Munich, Germany (e-mail: randolf.mock@siemens.com).

Corresponding author: difelice@cs.unibo.it

## I. INTRODUCTION

Nowadays, many governments of the world have identified Electric Vehicles (EVs) as a cornerstone toward a new generation of transportation systems able to reduce the pollution emissions while overcoming the dependence from fossil fuels [28]. Although such benefits are expected to become significant only on the long-term<sup>2</sup>, the market share of EVs is destined to quickly expand in the short-term, as a consequence of the increasing variety of vehicle models introduced by the car manufacturers. According to the analysis shown in [29], from 2010 to 2012, the overall number of sales of EVs has increased worldwide of more than 200%, in correspondence with the doubling of EVs models widely available to customers. However, several challenges must still be addressed so that a large-scale uptake of electric mobility might take place. Beside the well-known issues related to the EV performance (e.g. battery life and capacity), aspects on how to deploy the charging infrastructure, and how to manage the charging operations, are perceived as crucial from both the EVs drivers and the power companies. From the one side, given the duration of charging operations (in the order of tens of minutes), and the limited capacity of EV batteries, the placement of the Electric Vehicle Supply Equipments (EVSEs) in the current scenario has a direct influence on the experience of each driver (for instance, in terms of average trip time [24]), and might also impact the market penetration of EVs [30]. On the other side, several recent studies have demonstrated the harmful impact on the smart grid caused by the simultaneous charging operations of a multitude of EVs on an urban scenario [26][27], and have underlined the need of coordination between EVs and EVSEs [19][26]. In order to address these issues, current research investigates the utilization of Information and Communication Technology (ICT) for smart grid and electric mobility scenarios [2-8]. This is the case, among the others, of the Internet of Energy for Electric Mobility (IoE) project [31][34], which comprises 40 partners from 10 European countries, and aims at developing hardware, software and middleware solutions to assist and coordinate the charging operations by enabling reservations from the EVs drivers. Given the costs of building large-scale test-beds, simulation constitutes the reference technique to assess the effectiveness of the ICT-based solutions for Electric Mobility (EM) scenarios before a real deployment. Nowadays, several vehicle simulators are available [11-17], and provide a fine-grained model characterization of the components of an EV (including the electric motor, the

<sup>2</sup> According to [28], the impacts on fuel and electricity consumption by road passenger transport would be negligible until 2020-2025.

mechanical transmissions, etc). Similarly, battery models have been extensively investigated to compute the discharging of the battery over time, and to reproduce EV dynamics in an accurate way [15][16][17]. However, none of these tools can be used to study complex dynamics of EM scenarios (like the impact of EVs to the smart grid, or the effectiveness of charging reservation policies), where it is required to model several interacting entities characterized by different time-scale behaviors (i.e. EVs, EVSEs, the electrical grid, the communication network). To this aim, the most pioneering works are [22][23][24].

The main contribution of this paper is the design and implementation of an integrated framework to perform large-scale simulations of Electric Mobility (EMs) scenarios. Differently from the existing tools, our framework allows reproducing –within the same environment– dynamic aspects of the EVs (like mobility and discharging), of the EVSEs (like charging operations), and of communication between the EVs and the EVSEs (like the reservation process). To this aim, the framework relies on a co-simulation approach [3], which includes models deployed in SUMO [18], a microscopic traffic simulator, and OMNET++ [32], an event-based simulator, connected through the TRACI interface [33]. In SUMO, we model the characteristics of the urban environment (in our case, the city of Bologna), including the topology and altimetry information, and we generate the traffic load on the basis of macroscopic demographical data (i.e number of inhabitants, work places density, etc). In Omnet++, we implement models of EVs and EVSEs and we validate them by comparing the simulation traces with real consumption data of target EVs. Together with the mobility aspects, the Omnet++ modules provide the modeling of the data communication process between each EV/EVSE and a City Service (CS) that coordinates the charging operations over the scenario. All the simulated entities produce semantic data according to a smart-grid ontology [34] that guarantees interoperability among heterogeneous stakeholders (i.e. different EVs models, EVSEs providers, etc). Semantic data are collected into a shared repository called SIB (Semantic Information Broker) [42]. Our simulation tool can be used for two different analysis of EM scenarios:

- *Mobile Services evaluation and validation.* The simulation platform supports immersive emulation and validation of mobile applications related to the EM scenario (e.g. battery monitoring applications like [35][36]) which can be deployed in the near future to assist EVs drivers' operations. The communication between the real and synthetic world is bidirectional and works in real-time, i.e. the mobile applications can access the simulation data produced by simulated entities, and can also generate new events to be scheduled within the simulator (e.g. a re-route event of a vehicle). This is realized through a software sandbox (called Mobile Application Zoo) that implements the data exchange between the simulator and the mobile applications, while hiding all the details of the simulated world. As a result, mobile applications can be tested on a synthetic environment that reproduces most of the characteristics of a real scenario, with few or no changes with respect to their future deployment.

- *Pre-deployment analysis.* Through the modeling of EVs operations over realistic road topologies and vehicular traffic conditions, we are able to carry out quantitative analysis that can help the deployment of the charging infrastructure on a target scenario. Moreover, we can evaluate and compare different charging control strategies and reservation policies, and provide useful feedbacks about the impact of different EVs loads and drivers' behaviors on the charging service, and on the smart grid.

The rest of the paper is organized as follows. In Section II we review the existing literature about modeling and simulation of EVs and related scenarios. In Section III we present the EM scenario considered in this study. The components of the simulation framework are reviewed in Section IV. Model validation is discussed in Section V. In Section VI, we provide the details of the IoE architecture, and we describe how the simulator can be adapted to support immersive emulation of mobile applications for EM scenarios; to this aim, a charging reservation service is presented. In Section VII, we discuss how the simulator can be fed to perform pre-deployment analysis, and we provide results for a target scenario (i.e. the city of Bologna). Conclusions and open issues follow in Section VIII.

## II. RELATED WORKS

In this Section, we review the literature on simulation models and tools for Electric Mobility (EM) scenarios. We first analyze (in Section II.A) existing smart-grid simulators, integrating also ICT-related aspects. Then, in Section II.B we focus on tools to simulate the EV components, and the aggregate behavior of EV fleets.

### A. Models and Tools for Smart Grid Simulation

Although several definitions of smart grid have been proposed in the literature, all visions agree on the key role played by the Information and Communication Technologies (ICT) in improving the reliability and the efficiency of the electricity system [1]. At the same time, recent studies warn about the harmful impact produced by the communication delay on the smart grid dynamics [2][4]. In order to evaluate benefits and risks of the integration, novel modeling tools have been deployed to co-simulate the behavior of the smart grid [44] and of the communication network infrastructure. In [3], the authors review the existing co-simulation frameworks, and discuss possible approaches to guarantee tight synchronization among smart grid simulators (which are normally time-driven), and network simulators (which are normally event-driven). In [2], the authors describe how to integrate the Network Simulator 2 (NS2) with the Positive Sequence Load Flow (PSLF) grid simulator in order to test collaborative relay protection systems. Similar experiments are described in [4] and [5]. Other primary benefits provided by ICT solutions on smart-grid environments are the bidirectional metering with the possibility to control the self-production of energy from renewable sources (e.g. photovoltaic panels) and the interaction with the grid [6]. To this purpose, in [7] the authors describe an Omnet++/Matlab co-simulation framework through which they model a residential neighborhood scenario where some households have a photovoltaic panel installed, and investigate the impact on the distribution grid. A similar scenario is also described in [8] where the authors consider a dynamic price simulator

that takes into account live (residential) renewable energy sources, weather forecast and energy distribution load profiles. In [9] the authors utilize the CLEVER simulator to address a “what-if” analysis of smart metering and smart-grid systems in a large-scale urban scenario. Most of the co-simulation approaches described so far are based on monolithic simulation tools (e.g. NS-2) and thus might suffer from scalability problems when considering complex city-wide smart grid environments. To address this issue, a distributed platform simulator framework is described in [11]. Here, the smart grid is partitioned into different clusters. Each cluster can be simulated on a separate process, while a centralized component (called GridSpice) synchronizes the boundary states of the distributed processes that might belong to different tools [11].

### B. Models and Tools for Electric Vehicles Simulation

The intrinsic complexity of building large-scale EV test-beds makes the simulation a fundamental instrument to evaluate the performance of the target of interest. As a result, several modeling and simulation tools have been proposed in the literature [11-27]. These works can be classified based on the focus of the modeling phase into: (i) tools to simulate EVs components [11-17], (ii) tools to simulate EVs scenarios [18-25] and (iii) tools to simulate EV services [26-27]. The first class of tools provides a fine-grain characterization of the performance of an EV by considering all components (e.g. electric motors, batteries, etc) that are involved in the vehicle design cycle. Two vehicle simulation approaches [11] are possible: (i) backward facing and (ii) forward facing. In the first case, a speed trace produced by a real EV is used to compute the required torque, and then to derive the requirements of each electric motor component. No driver model is used. In the second case, a synthetic driver model is introduced to simulate the throttle and brake commands, and the resulting acceleration is computed on the basis of the tractive power produced by the electric motor, according to the vehicle (e.g. weight) and battery (e.g. efficiency) characteristics [11]. To this aim [12] reviews the existing relationships among the motor power, the vehicle/battery weights, and the maximum acceleration achievable by an EV, and proposes a set of MATLAB models to investigate the trade-offs among the design parameters of the vehicle. In [13] the authors describe the modeling and simulation (through MATLAB/Simulink) of an urban EV. A forward-facing model of both EVs and hybrid vehicles is proposed in [14] through the PSIM tool. A key component to correctly simulate the behavior of a EV is the battery model [15][16][17]. To this purpose, electrochemical [15][16] and electrical [17] models have been proposed. In [17] the authors describe Simulink models for three battery types (lithium-ion, nickel-metal hybrid and lead-acid), and then validate the predicted battery State Of Charge (SOC) against real traces. All the works mentioned so far provide a fine-grained characterization of the main components of an EV, but do not address the aggregated effect produced by a fleet of EVs moving on a city scenario. To this aim, several simulation tools [18-25] for EVs scenarios have been proposed by extending vehicular traffic simulators (e.g. SUMO [18]) or by integrating communication network, vehicular mobility and smart grid simulators. In [19], the authors analyze the voltage profile, the peak demand and the system losses caused on the grid by a varying number of

EVs in the Danish island of Bernholm, and conclude that controlling the EVs charging operations (e.g. performing them during off-peak hours) can accommodate more charging requests than a completely uncontrolled mode. Similarly, in [20] a co-simulation platform based on the Omnet++ and OpenDSS tools is used to study how to achieve coordinated charging of plug-in EVs through a Fiber-wireless communication infrastructure. An agent-based simulator of a smart-grid is described in [21], which models the energy production from power stations and the energy consumption of households (composed of different appliances) and EVs moving according to a random waypoint mobility model. However, in order to correctly estimate the battery consumption of EVs and their impact on the smart grid, realistic vehicular mobility models are required. For these reasons, several recent works propose SUMO extensions [22][23] that address the charging and discharging operations of EVs in city-wide environments, by considering the road topologies imported from OpenStreetMap [37]. In [22] the authors consider a scenario where EV drivers move from one parking spot to another one as a part of their daily activities, and study the possibility to deploy charging stations at each parking spot. However, charging and consumption models appear oversimplified, and no charging reservation process is evaluated. In [23] the authors extend SUMO by modeling the mechanical and electrical dynamics of an EV and by including altitude information that allows taking into account the discharge of an EV uphill, and the regenerative breaking downhill. In [24] a similar SUMO extension is described to simulate the operations of EVs over a realistic city map scenario (Vienna): a genetic algorithm is proposed to deploy the charging stations in order to minimize the vehicle trip time. In [25] energy-aware routing strategies for EVs are proposed and evaluated by modeling the traffic dynamics through a Markovian model. Finally, the last set of simulation tools [26-27] allows the modeling of communication services required to coordinate the charging operations of EVs on a large-scale urban scenario. In [26] the authors propose a load coordination protocol that allows an EV to notify its charging needs to a Load Coordinator entity. In [27] a simulator based on the Smart Objects System is proposed for EV parking spots, and different charging control policies are compared on the basis of how many EVs will leave the car park with the requested amount of energy in their batteries.

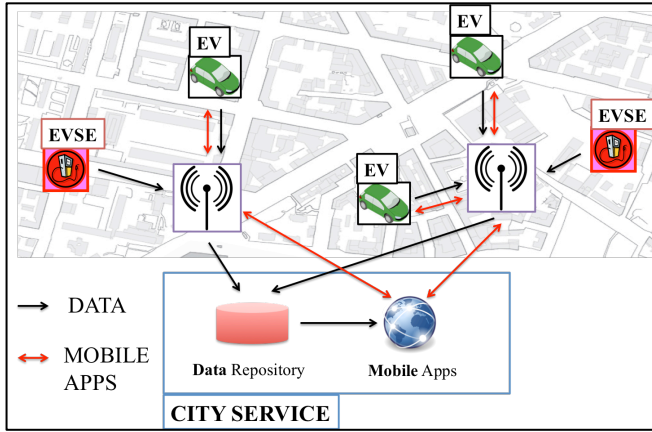
### C. Novel contributions of our work

For the goals of our study the most similar works are [22][23][24]. However, our study adds the following novel contributions:

- We model the operations of both EVs and of EVSEs (the latter are not considered in [22][23]). Leveraging the run-time information on the availability of EVSEs, we are able to model the EVs reservation process, and we can thus perform quantitative analysis about the effectiveness of existing charging infrastructures when considering the EVs load requests generated over realistic scenarios.
- We represent the data sharing process among the simulated entities (EVs/EVSEs) through a semantic ontology (defined within the IoE project). Moreover, we provide facilities for the real-time

immersive emulation and validation of mobile applications built on the same ontology, enabling the bi-directional communication between simulated and real components of an EM scenario.

- We build a co-simulation framework composed of Omnet++ and SUMO, through which we are able to model both vehicular mobility and communication between EVs and a stationary infrastructure.



**Figure 1.** The EM scenario considered in this paper.

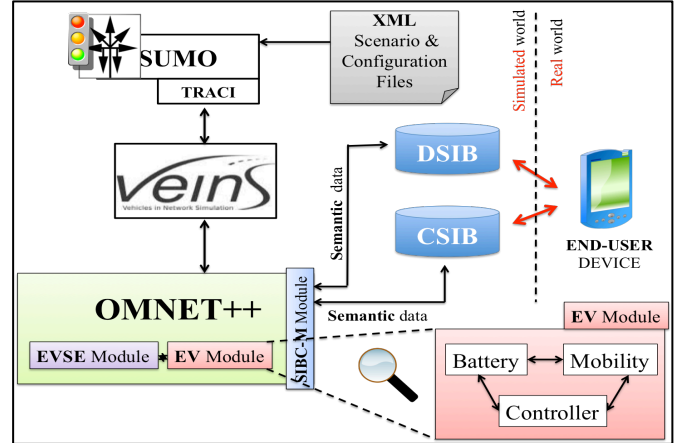
### III. THE ELECTRIC MOBILITY (EM) SCENARIO

From the previous discussion, it becomes evident that a smart-grid is a complex system consisting of several interacting agents. Developing a simulation platform modeling all possible scenarios and capturing all the system dynamics involved (including, for instance, dynamic energy pricing, impact of renewable sources, etc.) is a highly challenging task and it is far away from being addressed by the existing tools [1-11]. Therefore we restrict our focus on Electric Mobility (EM) scenarios, and we propose here a tool that allows the modeling of mobility, communication and coordination of EVs on large-scale, realistic urban environments.

Figure 1 shows the three main actors involved in our modeling, i.e.:

- EVs: they move inside the scenario based on drivers' specific habits and needs, and arbitrarily perform charging operations at the charging stations (EVSEs).
- EVSEs: they are stationary stations where EVs can stop and recharge of the requested amount of energy; EVSEs are connected to the smart-grid.
- City Service (CS): this represents a city-wide ICT infrastructure offering Internet services to the EVs and to the EVSEs providers, with the goal of coordinating the operations of vehicles' drivers and of the smart-grid. The ICT infrastructure is recognized as a fundamental component for the large-scale uptake of EM, and several architectures have been proposed in the literature [34][35][36][40]. In our case, without loss of generality, we assume that a CS works as a collector of information produced from EVs and EVSEs, which are then processed in order to provide Internet services to the EVs drivers by mobile applications. An example of such

applications is the charging reservation service that can greatly contribute to mitigate the risks of uncontrolled peak demands to the smart grid, as discussed in [19][22]. All the communication between CS, EVSEs and EVs is assumed to happen on a cellular (3G/4G/LTE) connection, although other wireless access technologies (e.g. Wi-Fi) might be also available in an urban environment and possibly included into our simulation platform.



**Figure 2.** The components of the IoE simulation framework.

### IV. THE INTEGRATED SIMULATION PLATFORM FOR EM SCENARIOS

In this Section, we provide a detailed overview of the simulation platform developed to evaluate EVs operations and services, based on the system model shown in Figure 1. Like previous works on co-simulation [2-9], our framework integrates multiple discrete-event simulators (e.g. SUMO, OMNET++, VEINS), each modeling a different aspect of the urban vehicular scenario (e.g. mobility, wireless communication, etc). Moreover, we extend these tools with new models addressing the unique characteristics of the EVs (e.g. battery discharging and electric motor dynamics) and of the EVSEs (e.g. charging operations), and enabling the immersive emulation and validation of mobile services. Figure 2 shows the main components of our simulation framework and the interconnections among them.

- SUMO stands for Simulation of Urban Mobility [18], and it is an open-source discrete event simulator for vehicular environments that is widely adopted by researchers working on the simulation of EM scenarios [22][23][24]. It can reproduce traffic dynamics on realistic road networks (e.g. imported from OpenStreetMap [37]), by micro-simulating the behavior of roadside components (e.g. traffic lights) and of each vehicle. Moreover it exposes a socket interface called TRACI [33] through which it is possible to retrieve data about the current simulation flow and/or modify the state of simulated entities. In our framework SUMO generates the vehicles routes and governs their mobility (both EVs and not EVs).
- OMNET++ is a discrete-event simulator used to model a large set of scenarios, ranging from wireless networks (e.g. sensor or vehicular ad hoc networks) to smart grid and smart metering systems [38]. The OMNET++ inner structure is based on

the concept of “module”. Each module models a different entity and can be connected to other modules via virtual gates, which are an abstract way to represent communication links. We extended OMNET++ by implementing the models of an EV and of an EVSE. Moreover OMNET++ offers facilities to interconnect real components with simulated modules, via C++ libraries or external sockets. We leveraged these functionalities to interface the simulated entities to an external semantic repository (called Semantic Information Broker - SIB [42]), which can then be accessed by mobile applications for immersive emulation purposes.

- VEINS is a vehicular simulation framework proposed in [39] to connect Omnet++ and TRACI. It provides Omnet++ developers with a complete set of APIs to query the SUMO platform and to dynamically retrieve information about the state of SUMO simulated entities. Moreover, it implements mechanisms to synchronize SUMO and OMNET++ at the end of each time step (equal to 0.1 seconds).
- The SIB works as information repository of the current EM scenario. More specifically there are two different SIBs in our simulation framework: a City SIB (CSIB) containing city level information collected by the CS to provide aggregated mobile services, and a Dash SIB (DSIB), working as local repository of each EV (e.g. to store mobility and battery-related information). Information is represented through the smart-grid ontology developed within the IoE project [34] to facilitate interoperable communication between different stakeholders.
- The XML Scenario Files characterize the target EM scenario and include: (i) the road network which can be imported directly from OpenStreetMap, (ii) the altimetry data which are useful to correctly estimate the battery SOC changes uphill and downhill (iii) the demographical data which provide statistical information about the drivers’ population and are used to generate realistic synthetic traffic loads and (iv) the EVSE network, which provides information about each EVSE (location, maximum power delivered, number of connectors).
- The XML Configuration Files specify the simulation parameters, like the EV penetration rate, the battery characteristics of each EV (e.g. initial SOC), and the SOC threshold<sup>3</sup> under which an EV must go to recharge ( $SOC_{min}$ ).

The simulation workflow proceeds as follows. At simulation startup, the XML Scenario and Configuration files are loaded into SUMO and the routes of the simulated vehicles (both EVs and non EVs) are generated on the basis of the demographical data by using the ActivityGen utility of SUMO. At each simulation step SUMO determines the mobility features (location, speed, acceleration) of each vehicle. This information is read by OMNET++ to update the current SOC of each EV, according to the battery model

described in Section IV.B. If a vehicle has a SOC value below the pre-defined threshold ( $SOC_{min}$ ) then it must recharge at an EVSE. In our simulator recharging operations can be modeled in two ways:

- *coordinated mode*: in this case the EV issues a charging request to the City Service (CS) through a cellular connection, indicating the amount of energy required, the reservation time, and the driver’s preference (e.g. the nearest station or the less expensive one). The CS will then reply with the location of the EVSE that meets the driver’s requirements.
- *uncoordinated mode*: in this case the EV moves to the nearest EVSE and recharges based on FIFO rule. No reservation is issued.

In both cases the vehicle is re-routed to the EVSE location by modifying its destination in SUMO through the TRACI `change target` command. Also, at each time-step, OMNET++ updates the SOC of the EVs that are currently charging, and manages the state of the EVSEs. The battery and mobility information of each EV are periodically updated to the DSIB, while the EVSEs and the charging information are inserted into the CSIB, in both cases by using the SIB Communication Module (SIBC-M) developed in OMNET++. When doing emulation, external applications can access the simulation data structures contained into the SIBs, and eventually schedule new simulation events.

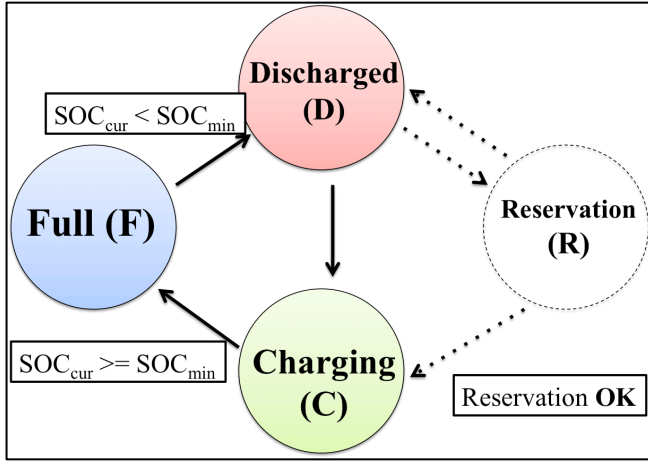
In the following the operations of the EV, EVSE and SIBC-M modules developed in OMNET++ are detailed.

#### A. The OMNET++ Module of an EV

The EV module models the aspects related to mobility, charging/discharging and routing of an EV. To this purpose, each EV is associated with a *state* variable, which describes its operating mode with respect to the current SOC value. In uncoordinated charging mode, at each simulated time-step an EV can be in one of the three states: Full (*F*), Discharged (*D*), and Charging (*C*). In *F* the EV has a current SOC value (i.e.  $SOC_{cur}$ ) higher than the threshold  $SOC_{min}$ , and thus moves inside the urban scenario according to the pre-defined routes computed at simulation startup time. When  $SOC_{cur}$  drops below the  $SOC_{min}$  (state *D*) the vehicle needs to recharge, and thus modifies its current path by approaching the nearest EVSE. Once at the EVSE, the vehicle stops and enters the *C* state until the battery is fully charged. At this point, the vehicle re-starts moving in *F* State according to its pre-defined route. Figure 3 shows the transitions among the EV states for both the uncoordinated and coordinated charging modes. In this latter case (shown through dotted lines), the Reservation (*R*) state is added. When a vehicle is discharged (*D*), it must contact the CS to reserve a charging slot, and then it can move to the selected EVSE. If no available EVSE is found, the vehicle performs a new reservation with modified preferences (e.g. reservation slot time). As shown in Figure 2, the OMNET++ module of an EV includes three sub-modules, respectively the Mobility, the Controller and the Battery modules. The Mobility module is responsible for querying periodically the VEINS framework, in order to obtain the current mobility parameters (location, speed, acceleration) of the simulated EV directly from SUMO. The Controller module manages the states’ updates of the EV, and thus its resulting behavior, on the basis of the current EV control mode. Two control modes are defined: (i) Simulated and (ii)

<sup>3</sup> This parameter models the “driver’s range anxiety” behavior described in [30].





**Figure 3.** The state diagram of an EV (in simulated mode).

User-Controlled. In the first case the Controller simply follows the state transitions defined in Figure 3. In the second case the behavior of the simulated EV is under the control of the end-user, through an external mobile application interacting with the Simulator in emulation mode. In this case, the user can arbitrary govern the transitions to the next states, i.e. a vehicle can decide to re-charge even if  $SOC_{cur} > SOC_{min}$ , thus jumping from state  $F$  to state  $C$ .

The Battery module is the main EV component and it is responsible for modeling the charging/discharging operations of the EV. More specifically, the consumption model governs the state transitions between the Full (F) and Discharged (D) state when the vehicle is moving. In Section II we mentioned the main techniques of EV battery modeling, based on electrochemical [15][16] and electrical [17] approaches. Despite their accuracy, these approaches might not to be suitable for large-scale simulation, due to the high computational load involved. Generally speaking, there exists a trade-off between system scalability and complexity of the consumption model, which must be considered with respect to the expected simulation output. In our scenario, we aim at obtaining city level statistics of a potentially high number of vehicles. For this reason, we attempted to limit the complexity of the consumption model while preserving as much as possible its accuracy for the sake of statistics reliability. The resulting model is based on macroscopic physical equations that derive the average power provided by the electric motor, and thus the battery consumption, based on the current vehicle dynamics. A similar approach is also followed in [13] and [23]. We assume that at each time step, the traction force  $F$  applied to each EV is the resultant of four components:

$$\vec{F} = \vec{F}_{ma} + \vec{F}_{mg} + \vec{F}_{drag} + \vec{F}_{roll} \quad (1)$$

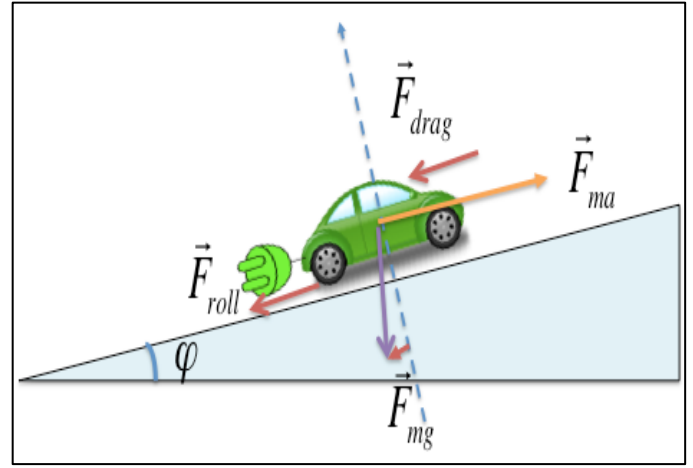
where  $F_{ma}$  represents the inertia of the vehicle,  $F_{mg}$  is the weight component along the  $x$ -axis,  $F_{drag}$  is the air friction and  $F_{roll}$  is the rolling friction. According to the laws of classical mechanics, these components can be expressed as follows:

$$F_{ma} = m \cdot \frac{v_t - v_{t-1}}{t_{step}} \quad (2)$$

$$F_{mg} = m \cdot g \cdot \sin(\varphi) \quad (3)$$

$$F_{roll} = m \cdot g \cdot \cos(\varphi) \cdot C_{roll} \quad (4)$$

$$F_{drag} = 0.5 \cdot \rho_{air} \cdot C_{drag} \cdot A_{cross} \cdot v_t^2 \quad (5)$$



**Figure 4.** The macroscopic physical model of an EV (in simulated mode).

Here,  $v_t$  and  $v_{t-1}$  represents respectively the vehicle speed at time instant  $t$  and  $t-1$ ,  $m$  is the mass of the EV,  $t_{step}$  is the simulation time-step,  $\varphi$  is the current slop of the road, and  $C_{roll}$ ,  $C_{drag}$ ,  $\rho_{air}$  and  $A_{cross}$  are parameters of the friction models (respectively equal to 0.015, 0.31, 1.21 and 0.16 in our experiments). We highlight the fact that –using the inclination angle (e.g.  $\varphi$ )– our model is able to capture the impact of sloped roads on the energy consumption of an EV. Figure 4 shows the components and the resulting force  $F$  acting on a target EV.

Based on the resulting force  $F$ , and on the current speed  $v_t$ , the average power provided by the electric motor for each simulation step ( $P_{mean}$ ) can be computed as follows:

$$P_{mean} = \frac{v_t + v_{t-1}}{2} \cdot F \quad (6)$$

Finally, the current SOC is estimated by removing from the residual battery charge the percentage of the total battery capacity  $B_{capacity}$  consumed in one time step (i.e.  $P_{mean} \cdot t_{step}$ ), further divided by the efficiency ( $\eta$ ) of the transformation from electric energy to mechanical energy. Therefore:

$$SOC_t = SOC_{t-1} - \frac{P_{mean} \cdot t_{step}}{B_{capacity}} \cdot \frac{1}{\eta} \quad (7)$$

The battery charging model governs the transition between  $C$  and  $F$  states. Differently from existing works that assume a linear increase of SOC while charging [24], for the computation of the new SOC our model takes into account the specific battery charger profile at each time step. The battery charging profile is modeled through a coefficients lookup table that alters the SOC increase based on the current charging phase:

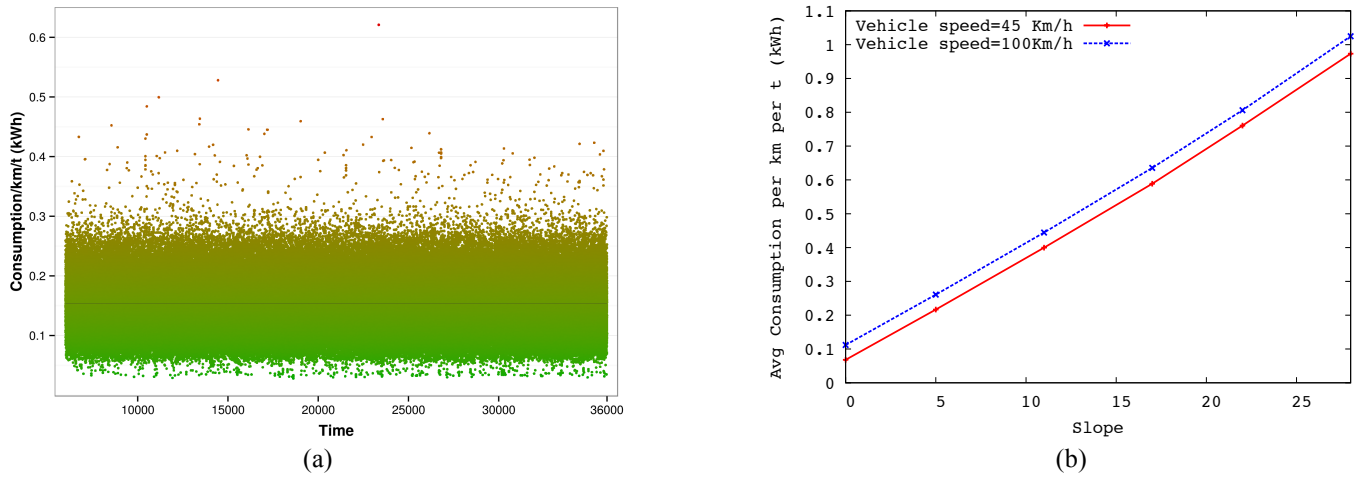
$$SOC_t = SOC_{t-1} + \frac{(pow(t) \cdot t_{step}) \cdot w(SOC_{t-1})}{B_{capacity}} \quad (8)$$

Here  $pow(t)$  is the power that can be delivered by the EVSE to the EV at time  $t$  and  $w(SOC_{t-1})$  is the coefficient assumed at the current state ( $SOC_{t-1}$ ). The coefficient values can be determined according to the charging characteristics of the battery (e.g. Li-ion) [26].

#### B. The OMNET++ Module of an EVSE

The EVSE module models the operations of a charging station in the addressed EM scenario. Each EVSE is characterized by the following set of parameters (written





**Figure 5.** Model validation (a) and parameter characterization (b).

into an XML scenario file): (i) the location, specified by 3D coordinates in SUMO, (ii) the maximum power ( $max_{pw}$ ) that can be delivered and (iii) the number of connectors ( $max_{conn}$ ) i.e. the maximum number of vehicles that can recharge in parallel at the same EVSE. Furthermore each EVSE has an internal state  $num_{conn}$  that accounts for the number of vehicles currently charging ( $num_{conn} \leq max_{conn}$ ). Each EVSE is also associated with a queue of vehicles that are waiting to access the charging service, in case this number is higher than the current availability indicated by  $max_{conn}$ . As already mentioned, in uncoordinated mode the vehicles are served according to a FIFO policy. In this case whenever a vehicle completes its charging process the EVSE sends a message to the EV module on top of the queue, which thus enters the  $C$  state. Vice versa, in coordinated mode, upon the arrival of an EV asking for a recharge, the EVSE goes through the following steps:

1. It checks if the vehicle has a scheduled recharging request.
2. If this is true, it acknowledges the request and starts the recharging session.
3. if the vehicle has not a scheduled recharging request and the EVSE is free, the EVSE asks the vehicle to make a quick reservation and then starts the recharging process.
4. Eventually, if there is a pending reservation made by another vehicle, the EVSE immediately tells the just arrived vehicle to vacate its current spot.

We highlight the fact that in our implementation the communication process between the EVSE and the EV is defined by the OMNET++ gate between the two modules, and here our reference model is the ISO 15118 protocol [41].

### C. The OMNET++ Module of the SIBC-M

The SIBC-M module enables the bi-directional exchange of information between the OMNET++ modules and the SIBs. This information is conforming with the EM ontology defined within the IoE project [34]. The data exchange with the CSIB models the communication between the EVs, the EVSEs and the CS according to the EM scenario described in Section III (Figure 1). As communication is expected to happen on a cellular connection we model the transmission

delay at each end point through the  $tx_{del}$  parameter. Vice versa, no communication delay is considered for the data exchange involving the DSIB as this models the communication inside each EV (e.g. to retrieve the battery SOC).

## V. MODEL VALIDATION

Correctness and reliability of output results produced by our simulation framework rely on sufficiently accurate models for car batteries and EVSEs, immersed in well-defined and plausible scenarios. Here, we discuss the model validation step, which in our case cycles between two phases: tuning of the models and comparison with real data. About the first aspect, we highlight the fact that EV models were not initially present in SUMO, and that using the generic vehicle profile provided by SUMO might lead to inconsistencies (for instance, about the weight, the maximum acceleration or the maximum torque of the electrical motor) that may affect considerably the realism of the simulation. For this reason, we created an extensible set of EVs with verisimilar mixture of parameters, from which the initial population of vehicles can be generated with the wanted proportion between the different profiles. About the second aspect, our issue was to investigate the accuracy of consumption results produced by the EV simulation model described in Section IV.A, when compared with consumption datasheet of real EVs. Generally speaking, energy consumption can be affected by several factors, including the traffic conditions and the specific driving style. For these reasons, the validation between simulation and real data can be performed only on *average*, when considering a sufficient number of EM scenarios and driving conditions.

In Figure 5(a), we depict the energy consumption experienced by a simulated EV, which moves within the Bologna scenario (further described in Section VII). The EV profile (weight: 2,587 ton, length: 5,0 meter,  $B_{capacity}$ : 40 kWh,  $\eta$ : 0,8) is based on the characteristics of a real EV prototype used for tests within the IoE project. A 24 hours simulation is considered. Each point of Figure 5(a) is the result of a single simulation, and represents the energy consumed by a vehicle on its random trip within Bologna; the energy values are normalized per ton and per Km, i.e. the energy consumed is divided over the weight of the vehicle and over the length of the path and is expressed in

**Table 1.** Average consumption of real and simulated EVs.

EV Class	Weight (ton)	Length (meter)	Battery Capacity (kWh)	Avg consumption (simulation) (kWh/(ton*Km))	Avg consumption (real) (kWh/(ton*Km))
Small	0.89	2.7	17	0.12	0.13
Medium	1.54	4.6	22	0.13	0.14
Large	2.58	5	40	0.15	0.15

in kWh/(km\*ton). The simulation is long enough to test different road traffic conditions, and different altimetry profiles of the paths chosen by the EV. In Table 1, we report the normalized average consumption of a simulated EVs<sup>4</sup> and real EVs<sup>5</sup>, for three different models of vehicles, whose characteristics (weight, battery, length) have also been reported in the Table. We can easily see that values in the two columns are quite close, thus validating the accuracy of the consumption model described in Section IV.A and of the car following model used by SUMO. In Figures 5(b) we provide further results that show how the proposed simulation model is able to characterize the consumption of an EV, on the basis of different driving style, path characteristics and vehicle profile. More specifically, Figure 5(b) shows the average consumption as a function of the slope of the road (a single straight road of 200 Km and variable density of vehicles are considered), for two different values of the EV speed (45 Km/h and 100 Km/h). It is easy to see that the path altimetry might have a significant impact on the consumption of the EV, thus justifying the deployment of mobile applications that take into account the altimetry of the path for residual SOC prediction [43].

## VI. FIRST USE CASE: TESTING AND VALIDATING THE IOE MOBILE SERVICE PLATFORM ON EM SCENARIOS

In this Section, we provide insights on how the proposed simulation framework can be used to test and validate Internet mobile applications related to the EM scenario. More specifically, our validation focuses on the IoE platform for mobile services, whose main characteristics are detailed in Section VI.A. In Section VI.B, we describe how it is possible to embed mobile Android applications into the simulation platform of Section IV, using the sandbox offered by the Mobile Application Zoo (MAZ). Finally, in Section VI.C, we provide an example of an Android application we developed to assist EV drivers with charging operations.

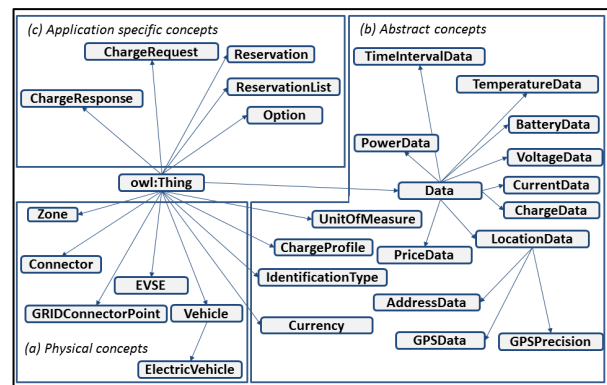
### A. The IoE platform of Internet services for EM scenarios

One of the key challenges of EM scenarios is the intrinsic heterogeneity of domains, platforms and stakeholders, which might pose several issues to the provisioning of city-wide Internet services for EV drivers. In order to tackle such heterogeneity, and to guarantee data interoperability among different producers and consumers, the IoE architecture relies on the Smart-M3 technology, the main characteristics of which are briefly reported below.

<sup>4</sup> The value for the EV (Large) is obtained by averaging the points of Figure 5(a).

<sup>5</sup> For the Large class, the value is derived by measurements we obtained on repeated driving scenarios with the same target vehicle. For small/medium class, the value is obtained from technical datasheets released by the cars' manufacturers.

In the Smart-M3 platform, information is represented as a set of semantic triples corresponding to a directed labeled graph, whose nodes and arcs are univocally identified by URIs. The semantic graph is based on ontologies i.e. machine interpretable domain descriptions, hosted by a Semantic Information Broker (SIB). The SIB stores an RDF (Resource Description Framework) graph exposing primitives to modify or query it through the mostly known standards i.e. SPARQL query language or RDF/XML information format. The SIB also supports the subscribe-notify mechanism to reduce traffic by eliminating polling queries and to improve the user experience on mobile services through reactivity to complex events. SIB performance and features have been described in literature [34][42] and are improving in time since it is an active open source project. Finally, the Smart-M3 foresees the deployment of Knowledge Processors (KPs), which are third-party software agents able to access the SIB and manipulate the semantic information. By referring to the EM scenario depicted in Figure 1, the CS repository is managed through a SIB instantiation (called C-SIB), while the mobile services are implemented through KPs performing data mining over the C-SIB. Relevant vehicle data are collected and sent to the C-SIB by the user smartphone. This may occur with or without the addition of a software component to the EV's on-board telematics unit. In the architecture considered in this study, we assume a local SIB (called D-SIB) being integrated into the EV on-board unit, although this element is not fundamental on a real deployment, as also discussed in [34]. Similarly, the EVSE can access the C-SIB by using adapter KPs that translate legacy information into RDF sub-graphs (this might allow, for instance, providing information about current availability and energy prices at each EVSE, although the latter are managed by different providers). The mutual understanding between all actors is enabled by a domain ontology [45], which is designed to capture most of EM concepts and of EM-related application requirements.



**Figure 6.** The ontology class tree for the EM domain.

The class taxonomy used by the IoE ontology is shown in Figure 6. It is divided into 3 areas depending on the nature of the concept: (i) physical entities (e.g. Vehicle, EVSE, Connector), (ii) abstract entities (e.g. Data, ChargeProfile) and (iii) service specific terminology (e.g. ChargeRequest, ChargeResponse, Reservation, and so on). The service specific classes are modular extensions of the core ontology and introduce concepts relevant to the EM applications. Besides the class taxonomy, the ontology includes properties to specify relationships between individuals and to describe their features in a machine interpretable way. When the information of the real world is mapped according to the ontology, the resulting RDF graph can be then explored by KPs using the SPARQL query language.

#### B. Mobile Application Zoo (MAZ): Embedding real components into simulated entities

In a real deployment of the IoE service platform, the C-SIB receives data from EVs through a wireless Internet connection, while the mobile applications running on users' smartphones invoke the KPs offered by the CS. Validation and testing of such architecture might be difficult to be performed due to the costs of setting up large-scale test-beds, and the current limited penetration of EVs in our city scenarios. The simulation platform described in Section V provides an ideal environment to test the scalability of the IoE service platform under synthetic loads of EVs, and to evaluate the impact of reproducible scenario characteristics (e.g. drivers' anxiety) on the system performance. To this purpose, we highlight that the simulation platform of Figure 2 integrates the C-SIB and D-SIB components without any implementation changes with respect to a real EM deployment. The only difference relies on the data population process, since in our case the SIBs are fed with data from simulated EM entities, rather than from real EVs or EVSEs. More specifically, the D-SIB is connected to each simulated EV, and stores its current position and battery related information (e.g. current SOC). The C-SIB is fed with EVSE status information, and keeps track of charging reservations requested by EVs. The communication between the simulation entities and the SIBs is managed through the SIBC-M Module previously introduced in Section IV.C. Another distinctive feature of our simulation framework is the possibility to embed mobile applications deployed over the IoE service platform into simulated EM environments. As a result, developers can test their applications over realistic EM scenarios in a seam-less way, and with no changes with respect to the deployment on a real scenario. We call this sandbox the Mobile Application Zoo (MAZ), in order to stress its generality with respect to the EM applications it is possible to validate. The operational behavior of the MAZ component is based on three design choices:

- *Adoption of a common ontology.* Both real and simulated components utilize the EM ontology previously described. As a result, the smartphone can communicate directly with the SIBs through its KP, and retrieve the information produced by the simulation. No direct communication is involved between the simulators (Omnnet++/SUMO) and the smartphone. As a consequence, no changes are required to the simulation models and to the mobile applications, since they are unaware of the presence of each other.

- *Real-time synchronization.* In order to accomplish realistic emulation, we use a specific feature of OMNET++ that allows the synchronization of simulation execution with real wall clock time (class `cRealTimeScheduler`). The clock synchronization is then extended from OMNET++ to SUMO through the VEINS component.
- *Geo-coordinates synchronization.* Nowadays, most of mobile applications include geo-localization capabilities, the relevance of which is even higher when considering the EM domain [35][36][40]. The MAZ sandbox allows to test geo-localization and route navigation functionalities, thanks to the facts that: (i) simulated EVs can move into realistic road map topologies imported from OpenStreetMap and (ii) the SIB-C Module is in charge of converting the position of each simulated entity from the internal coordinates (used by SUMO/OMNET++) to the latitude/longitude coordinates system used by Google Maps. In the SIBs, the localization coordinates are always represented in this latter format, so they can be directly displayed on the map of the smartphone.

The MAZ supports bi-directional exchange of information with the SIBs, corresponding to two different interaction modes between real and simulated components:

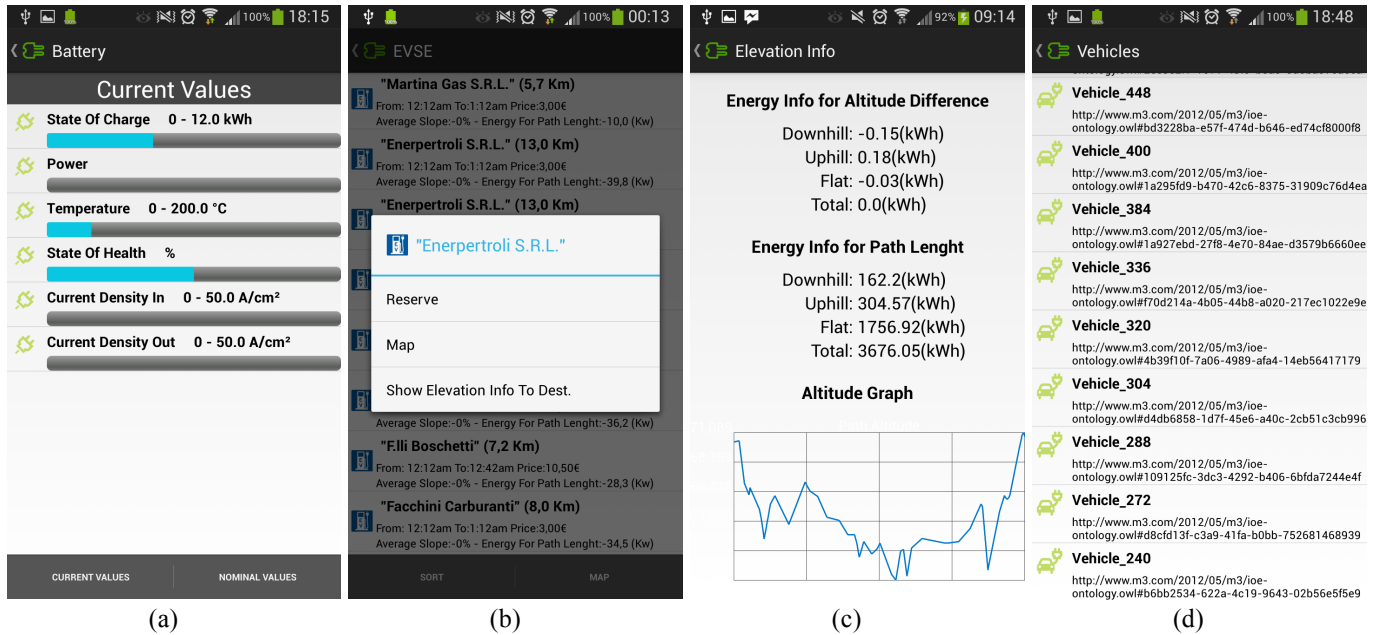
- 1) *Data retrieval.* The mobile application accesses the SIBs in read mode, and retrieves data produced by simulated entities, in order to consume them locally on the smartphone. This is the case of mobile applications that display EV vehicle data (e.g. battery-related information), or context-related information (e.g. closest EVSE).
- 2) *Data insert.* The mobile application accesses the SIBs in write mode, and can thus insert new RDF triples generated at run-time as a consequence of some actions performed by the user interacting with the smartphone. Such data might then cause the schedule of new simulation events, which are dynamically added to the event queue used by OMNET++, and can thus modify in real-time the execution flow of the simulation. As an example, a user might issue a charging reservation through the mobile application, which in turn causes a state change of the controlled simulated EV in OMNET++, which in turn fires a re-routing event in SUMO.

In the following, we provide an example of IoE mobile applications that have been tested through immersive emulation in our simulation framework.

#### C. An example of mobile application embedding

As proof of concept of the MAZ system, we developed a mobile Android application that offers multi-level assistance to EV drivers, and we tested it on a synthetic EM scenario modeled through our simulator. More specifically, our application was designed to support these three main functionalities:

- 1) Visualization of battery-related information, e.g. the current SOC (screenshot in Figure 7(a)).
- 2) Dynamic reservation of charging slots at available EVSE (screenshot in Figure 7(b)).



**Figure 7.** Screenshots of the mobile application for SOC monitoring and on-line charging reservation.

- 3) Visualization of path-related information, e.g. current altimetry profile, and available EVSE (screenshot in Figure 7(c)).

In a real deployment, the mobile application can access the battery-related information stored in the on-board D-SIB of the EV through a Bluetooth connection. Information about the path and about the EVSEs is provided by the C-SIB, which in turn is accessed through a cellular Internet connection. The reservation process follows the communication protocol previously described in [34]. First, the user requests a recharge operation to the CS attaching additional information like its current position and recharge preferences (i.e. closest EVSE, EVSE with minimum queuing time, EVSE with lowest energy price). The CS analyzes the user request, computes the best available options over the whole city context based on the current status of the EVSE (this data are available on the C-SIB) and provides the processed alternatives. Once the user has received the list of EVSEs, s/he can decide the most suitable one and waits for the final confirmation, which is mandatory in order to avoid synchronization issues due to the concurrency of user requests. In order to connect the application to the simulation framework through the MAZ, we introduced only a slight change to the mobile application, i.e. we added a new Android activity, which is displayed only when the simulation is running. Through this activity (shown in Figure 7(d)), the user can decide which EV to control, among the list of currently simulated EVs. From that moment, the selected EV enters in Controlled mode, and the user is immersed in the simulation as s/he was driving that vehicle. For instance, s/he can see the update of battery-related information in his virtual on-board unit (Figure 7(a)), and track its position displayed on the Google Maps. Also, the user can arbitrarily change the current state of the vehicle by issuing a reservation request. In that case, s/he can see on the map that the vehicle will change its route and stop to the selected EVSE, in order to recharge the desired amount of energy. Through the embedding described so far, we validated the correctness of

applications described in [47,48] before their real deployment.

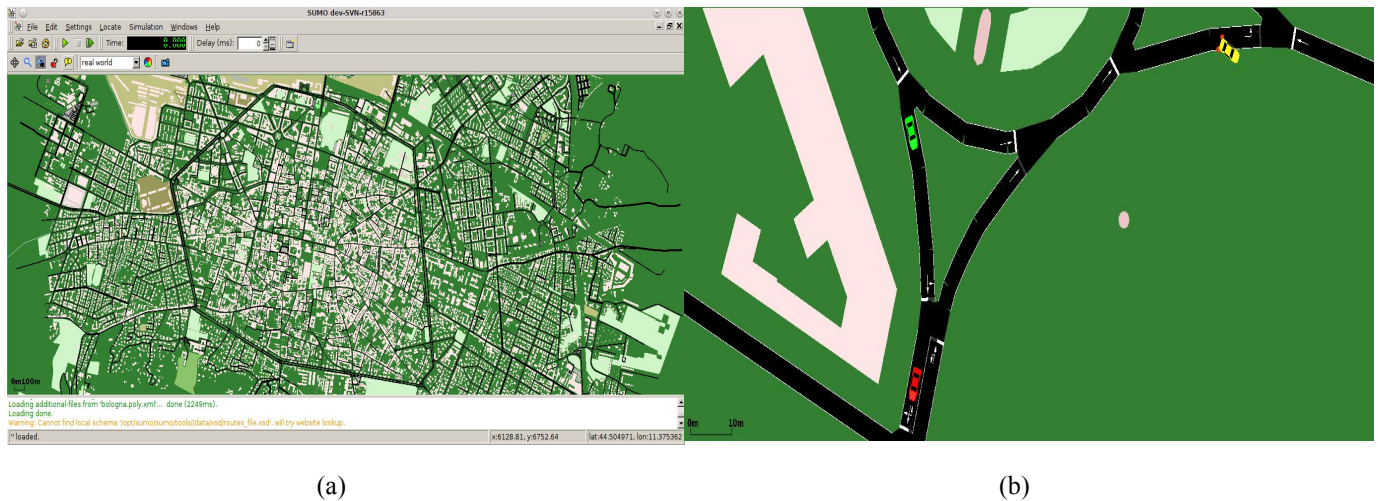
## VII. SECOND USE CASE: PRE-DEPLOYMENT ANALYSIS OF EM SCENARIOS

In this Section, we provide results that demonstrate the potential of our simulation framework to provide quantitative analysis of charging infrastructures and services over realistic large-scale EM environments. To this purpose, in Section VII.A we first introduce the modeling of the scenario (i.e. Bologna), by using realistic data about EVSE positions, power characteristics and vehicular traffic flows. Then, in Section VII.B we present a subset of results which can be extracted from the simulation platform, and which shows the impact of the charging reservation service on the efficiency and occupancy of the charging infrastructures. Finally, in Section VII.C, we discuss a methodology to estimate the impact of charging operations performed by EVs on the target scenario, by using the output of our simulator (i.e. the EVSE load history) as input of a grid planning tool (e.g the Siemens PSS Sincal software [46]).

### A. Modeling of the EM scenario

To model the electrical mobility scenario, we focused on three different aspects: the vehicular road topology, the vehicular traffic flow, and the EVSE network. About the first aspect, we imported the road topology map of Bologna directly from OpenStreetMap [37], and we fixed the main imprecisions regarding the traffic lights network and, the directions allowed on each road (e.g. one-way or two-way). Regarding the traffic flows, we utilized real data statistics about the vehicles flows on the major roads of the Bologna city area, based on a measurement campaign performed by the Bologna municipality. We considered two classes of vehicles in our simulations: (i) EVs, modeled according to the methodology described in Section IV.A, and (ii) normal (fossil-fueled) vehicles, modeled according to the basic vehicle model provided by SUMO. Normal vehicles are added to the scenario in order to produce a synthetic background traffic, although the focus of the analysis is on





**Figure 8.** The SUMO screenshot of the simulated scenario (a) and a detail with three different vehicles (b).

the operations of the EVs. During each simulation, we kept constant the overall number of vehicles moving within the scenario, and the ratio between EVs and normal vehicles. Regarding the EVSE network, we considered the real map positions of the EVSEs, also available on the EnelDrive website [40]. At present, a total of 20 EVSE are present in the city of Bologna, and 13 of them are located in the city center, which are also the ones used in our simulations. Figure 8(a) and 8(b) depict two screenshots of the simulated scenario within SUMO. More specifically, Figure 8(a) shows the complete area, while Figure 8(b) shows a detail with three vehicles: the green one is a simulated EV, the yellow one is an EV doing charging operations, and the red one is a normal (fossil-fueled) vehicles.

The area, simulation, EVs and EVSEs parameters are collected in Table 2. Although we present here the results about the Bologna scenario, we also highlight that the IoE simulator can be easily tailored to evaluate and model any other EM scenario, once the correct tuning of the modeling parameters is performed. At present, this is being performed out for the German city of Erlangen, for which a comprehensive set of data from a measurement campaign is available, beside the all the details of the traffic infrastructure.

**Table 2.** Simulation parameters

Simulation Area size	7.2km x 8.6km
Number of EVs	20,40,100,200,400
EV weight	2572 kg
EV Battery capacity	20 kWh
Number of EVSEs	14
$max_{pw}$	20 kW
$num_{conn}$	1
$SOC_{min}$	10%, 20%, 30%
Simulation length	80000 seconds
$t_{step}$	0.1 seconds

### B. Simulation results

In this Section we present the results obtained through extensive simulations of the target scenario. More specifically, Figures 9-12 show the benefits provided by the IoE architecture in terms of EVSE scheduling and

management, in comparison to the uncoordinated charging scenario, where reservation based on mobile services is not utilized. In Figures 13-15 we provide insights about the ability of the charging infrastructure of the target scenario to accommodate the load requests of a varying population of EVs.

Figure 9 shows the percentage of time in which the EVSEs are occupied, while varying the number of EVs in the scenario. Three different configurations are considered:

- The red bar refers to the case in which the IoE architecture is fully deployed, and vehicles can reserve for a charging spot, according to their SOC values. The reservation look-ahead is fixed to 24 hours, i.e. an EV will check for a reservation available in the next 24 hours, starting from the three closest EVSEs.
- The green bar refers to the case in which no reservation is performed, and thus an EV that needs to recharge goes and checks to the closest EVSE. In case this latter is not available, the EV moves to the second closest EVSE, then eventually to the third, then again to the first, and keeps polling in this way till an available EVSE is found.
- The blue bar refers again to the IoE architecture with mobile reservation enabled, however a short look-ahead of 3 hours is used.

Figure 9 demonstrates that a short charging look-ahead translates into suboptimal utilization of the EVSEs. This happens because EVs can not schedule charging spots too much in advance, and thus lots of “charging holes” might appear between consecutive reservations, leading to a poor overall utilization. This effect might also be seen as a special instance of the well-known knapsack problem. On the opposite, in the other two configurations, the EVSEs are occupied according to the load requests, although utilizing reservation provides slightly lower utilization than the uncoordinated case. The explanation of this fact is that in our simulation model, EVs park themselves while waiting for their reserved charging spot. As a result, the EVSEs might not be fully utilized in low-density scenarios due to the impact of the traveling delay (i.e. the time required from

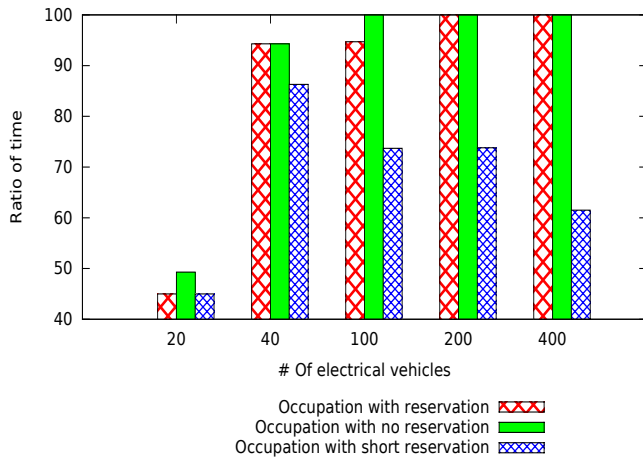


Figure 9. Percentage of EVSE occupation time.

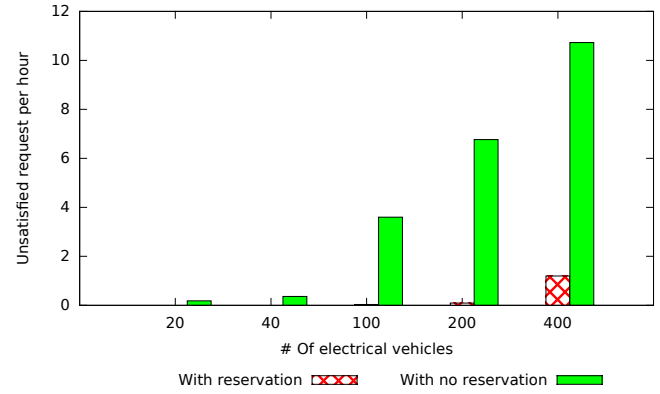


Figure 10. Unsatisfied charging request per hour.

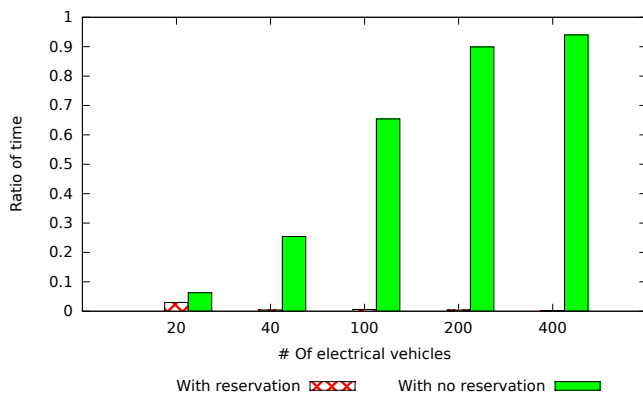


Figure 11. Ratio of time among EVs driving states.

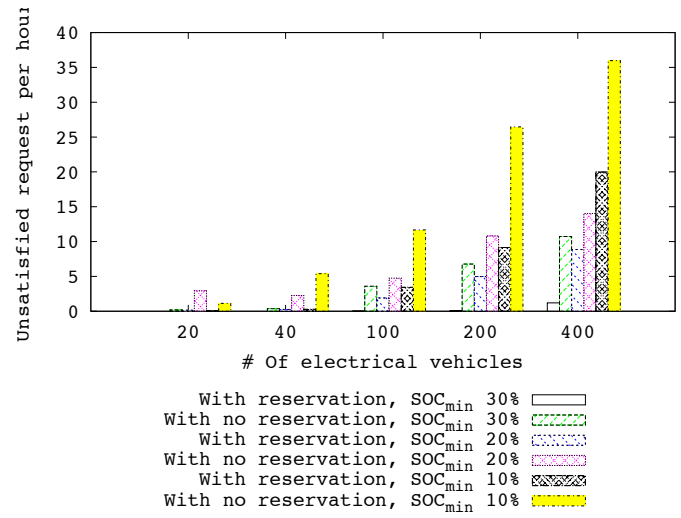


Figure 12. Unsatisfied request vs  $SOC_{min}$  values.

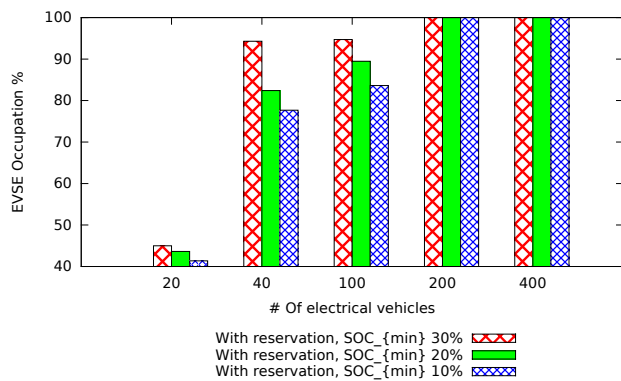
a parked EV to reach the EVSE where it performed the reservation). Such a delay is negligible when reservation is not used, since vehicles continuously move from an EVSE to another, till an available one is found. Thus, the probability of having empty slots among consecutive charging operations is reduced. From the considerations so far, a question might arise whether the reservation service provides a constructive, real benefit for the deployment of a charging infrastructure. In Figures 10 and 11 we add more results to address this issue.

More specifically, in Figure 10 we depict the number of unsatisfied requests per hour, both for the reservation case (look-ahead of 24h) and without reservation. An unsatisfied request is defined as the event in which an EV runs out of battery since all the EVSEs are busy, and no charging can be performed. Figure 10 shows clearly the benefits of the reservation service, mainly due to the intelligent scheduling and planning ahead. Of course, increasing the number of EVs still produces an increase of unsatisfied requests also for the reservation case, but in any case this value remains much lower than the uncoordinated case. From these results, two considerations can be argued. First, when the reservation service is used, accommodating the EVs load request translates directly into providing a sufficient number of EVSEs and into placing them in favorable spots, while these actions might not be enough when reservation is not

used. Second, since the reservation can guarantee the availability of the charging service from the perspective of the EV driver, the anxiety problem as mentioned in [30] might be mitigated, and the overall market of EVs could take benefit. In Figure 11, we further analyze these last issues, by analyzing the average ratio of time spent on different states from each EV. More specifically, we depict the ratio between the amount of time in which the EV is going to the EVSE and the amount of time in which the EV is doing its normal activity, like driving and moving toward to a Point of Interest (i.e.  $ratio = \frac{Going\_to\_recharge}{Driving}$ ).

Clearly, the reservation service provides the best results, since the EV reaches the EVSE only after a charging slot has been guaranteed by the CS. A direct consequence is that EVs drivers can have more time to perform their daily activities when compared to the case without reservation, when additional time and battery capacity might be required to discover an available EVSE. In Figure 12, we depict the unsatisfied request metric, for the two configurations (i.e. with and without reservation), while varying the  $SOC_{min}$  parameter, i.e. the threshold percentage of SOC below which an EV driver decides to perform a recharge. Varying the  $SOC_{min}$  parameter corresponds to model different anxiety levels from the EVs drivers, and thus different driving behaviors. Increasing the  $SOC_{min}$  value produces a



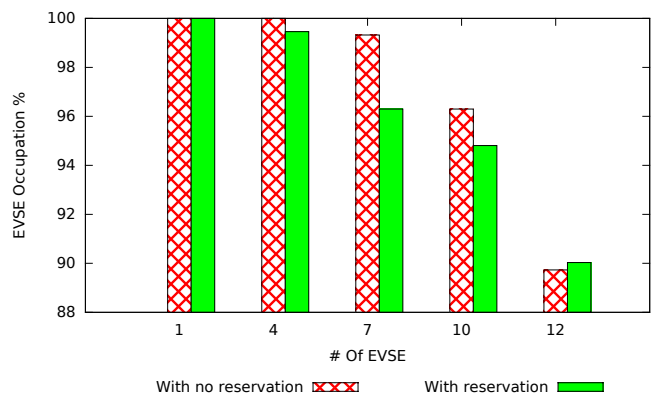


**Figure 13.** EVSE occupation time for different  $SOC_{min}$  values.

higher number of recharging operations performed in the scenario. However, Figure 12 shows an apparently counter-intuitive behavior, i.e. the number of unsatisfied requests is higher for lower anxiety values. This can be justified by considering that when the  $SOC_{min}$  threshold is low, the EVs might not have enough battery charge to reach the EVSE, in both the configurations. However, it is still evident that enabling the reservation might accommodate a higher number of requests from EVSEs, whatever the value  $SOC_{min}$  is.

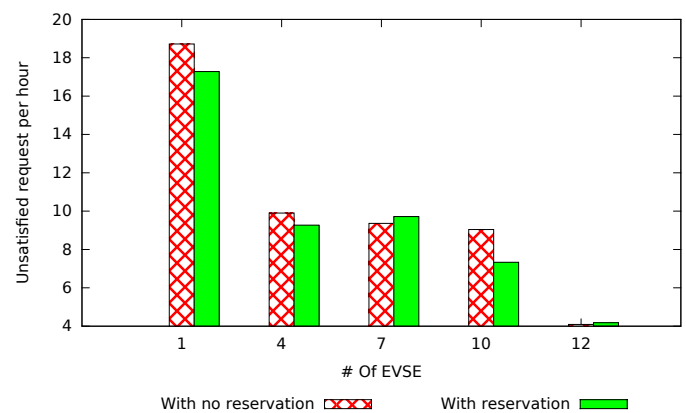
The presented results shown so far demonstrate that with the IoE architecture more charging requests can be accommodated, while also saving more time to the users for their daily activities. In the Figures below, we investigate the ability of the charging infrastructure to accommodate the load requests from a varying number of EVs, upon varying drivers' behavior and upon varying availability of the EVSEs. Reservation is always enabled, with look-ahead equal to 24 hours. More specifically, Figure 13 depicts the percentage of EVSE occupation time, when we vary the number of EVs and the values of  $SOC_{min}$ . Intuitively, the EVSE occupation time increases when more EVs move within the scenario, or when they need to recharge more frequently (i.e. for higher values of  $SOC_{min}$ ). Regardless of  $SOC_{min}$ , the saturation is achieved with 200 EVs, which points to the fact that for higher penetration rates of EVs the current charging infrastructure might not be enough to accommodate the load. In Figure 14 we further itemize this issue, in that we show the EVSE occupation rate by fixing the number of EVs (40), and varying the number of EVSEs active in the scenario, i.e. by assuming that only a fraction of the available ones for the Bologna scenario are actively working. In our experiments, EVSEs are progressively added to the scenario by considering their distance from the city center (closest ones first). As expected, the average occupation decreases while increasing the number of charging stations available in the scenario.

In Figure 15, we conclude the analysis, by showing the number of unsatisfied requests per hour when we vary the number of EVSEs available in the scenario, according to the methodology of infrastructure deployment previously introduced. Referring to the load produced by the 40 EVs, it is easy to see that a number of EVSEs lower than 10 might not be enough to accommodate all the requests originated by the EVs. Of course, the location of the EVSEs is also fundamental to compute the unsatisfied request ratio. In our analysis, we utilize the real locations of EVSEs, however we



**Figure 14.** EVSE occupation time for varying number of EVSEs.

highlight that our simulation framework might be also utilized as a planning tool to decide the optimal locations of charging stations [23].



**Figure 15.** Unsatisfied requests for varying number of EVSEs.

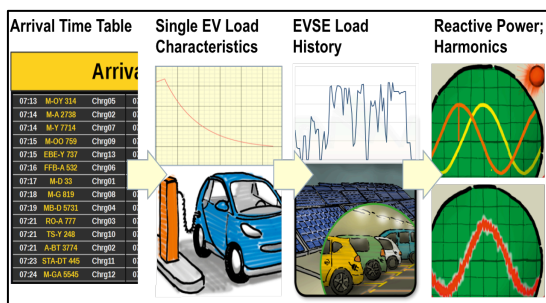
### C. Methodology to evaluate the impact on the smart grid

Beyond the simulation of EV/EVSE/mobile services that we have presented in depth in this paper, there is another valuable outcome of our simulation framework that is related to its integration with production-level grid planning tools in order to realistically estimate the grid impact of EVs moving on large-scale scenarios such as for example the city of Bologna. More specifically, in the proposed methodology the integrated SUMO/OMNET++ simulation platform can deliver EVSE load history data as input to the feeders of local distribution grids. To this purpose it is worth to highlight that being able to calculate the load of an EVSE is crucial for the planning of future decentralized grids because there is no significant history of EV recharging patterns so far. In fact, although several smart grid simulators have been proposed in the literature (reviewed in Section II), they might likely fail for not being able to generate realistic EVSE loads over time, since they do not model the characteristics of vehicular mobility, and thus they might not reproduce the effective recharging needs accurately. Conversely, these aspects are taken into account by our simulation framework, where we model the dynamics of *i*) vehicular congestion (through the car-following models provided by SUMO), *ii*) charging requests (through the battery model described in Section IV.A) and *iii*) vehicle arrivals and departures at each EVSE.

Based on these considerations, we propose the following

four-step methodology to compute the EVSE load history from simulation data produced by our simulation framework:

1. First, the arrival time-table of EVs at an EVSE is calculated for the specific target scenario (e.g. Bologna), through the modeling described in Section VII.A. EVSEs may consist of several chargers (e.g.  $num_{conn} > 1$ ), with different values of output power.
2. Knowing both the start time of a recharging session (from the arrival time table) and the charging power as a function of time (based on the  $w(SOC_t)$  coefficients introduced in Equation 8), the charging duration of a given EV/charger combination can be determined for a given set of initial and target SOC, which in turn can be either experimental data or parameters calculated according to detailed physical models and EV routes.
3. Under consideration of different reservation and charging strategies and while obeying given recovery times of the charging infrastructure, the time dependent load characteristics of an EVSE can thus be calculated and delivered in a format compatible with the requirements of a grid planning tool (e.g. for the Siemens PSS SINCAL software [46], the CVS or Excel format).
4. With the load history information available, the impact of an EVSE infrastructure in operation on the grid - i.e. in terms of load, reactive power and harmonics propagation - can be determined and thus allow the planning of future decentralized grids including EV recharging infrastructures (rightmost illustration in Figure 16).



**Figure 16.** Methodology to generate EVSEs load history traces.

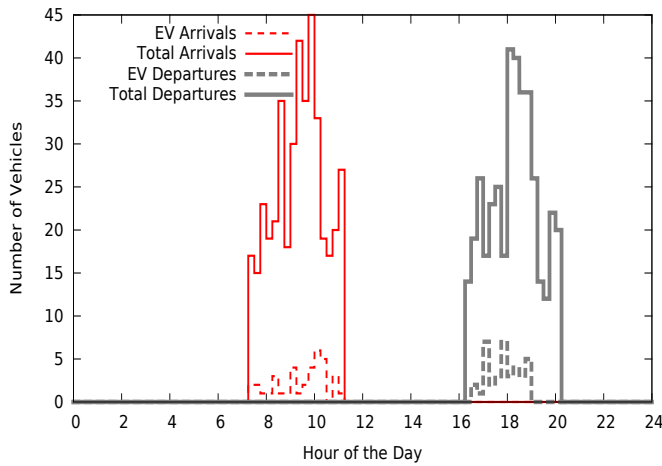
In the following a proof of concept of steps 1-3 is provided for a reference scenario. More specifically, a parking spot located near an office district of the Bologna scenario is considered. The parking spot is provided with an EVSE to allow the charging of parked EVs. In order to compute the arrival time table in a realistic way (step 1), we built an activity traffic model through the ActivityGen utility of SUMO [18]: we provided as input Bologna demographical data and the street topology (imported from OpenStreetMap), and we got as output the list of vehicles routes. We modeled the parking duration through a probabilistic distribution, based on typical office opening times (i.e. from 7am to 8pm). Figure 17 depicts the number of vehicles (conventional and EVs) in the parking spot over the 24 hours. All arrivals are concentrated between 7am and 11am (with a peak between 8am and 9am), while most of the departures occur around 6pm; this result confirms the correctness of the model implementation. Figure 18 shows

the total energy delivered by the EVSE to the parked vehicles, over the 24 hours. We consider here an EVSE with maximum power equal to 40 kW ( $max_{pw} = 40$ ) equally shared among its connectors, with each connector rated to a maximum of 10 kW, and we consider three different connector configurations (i.e.  $num_{conn}$  equal to 2, 3 and 4, respectively). Moreover, we test two different parking management schemes: (i) without reservation: in this case, EVs can recharge if they find a free connector once they enter the parking spot, otherwise they will park somewhere else and will not recharge. (ii) with reservation; in this case, EVs book a charging slot before their arrival at the parking spot, and at their reservation time, they get connected to the EVSE (for instance by a valet) and start recharging. Based on the simulation results Figure 18 summarizes recharging duration (step 2), and the time dependent load characteristics of the EVSE (step 3), which is saved in CSV format for grid impact evaluation through the PSS SINCAL [46] software (step 4).

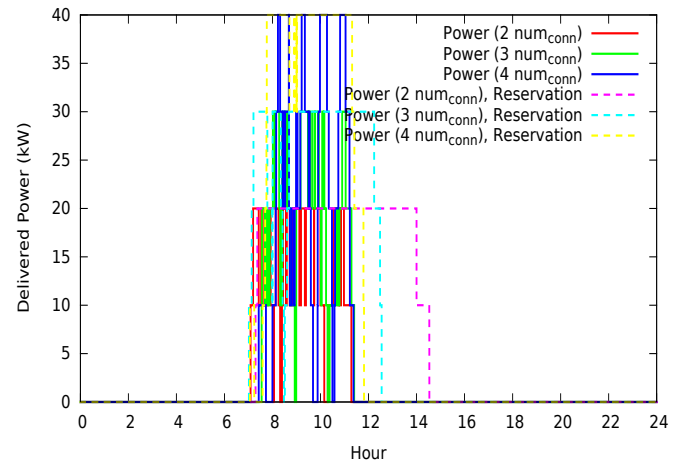
Figures 19 and 20 show additional results that demonstrate how our simulation platform can assist the pre-deployment analysis of the charging infrastructure on the target scenario, by providing useful indications about the correct EVSE sizing and configuration based on the load produced by the EVs. More specifically, Figure 19 depicts the ratio of fully charged EVs, i.e. of EVs that leave the parking spot with SOC equal to 100%, as a function of number of connectors ( $num_{conn}$ ) available at the EVSE. We test the two parking management schemes defined above (i.e. with reservation/without reservation), considering two different EVSE power profiles ( $max_{pw} = 50$  kW and  $max_{pw} = 100$  kW). The following considerations can be drawn from Figure 19: (i) increasing the power profile of the EVSE produces a poorer improvement than increasing the number of available connectors; (ii) when reservation is not available, at least 15 connectors (i.e.  $num_{conn} \geq 15$ ) are needed in order to serve all the EVs; (iii) when reservation is used, 5 connectors are enough (i.e.  $num_{conn} \geq 5$ ) to obtain the same result. Figure 20 compares the average EVSE occupation of the two parking management schemes, as a function of  $num_{conn}$  and for two values of  $max_{pw}$ . Here it is possible to notice that - for  $num_{conn} \leq 10$  - the occupation is much higher when the reservation service is used. This is reasonable as more EVs are served during the 24 hours, in fact even if a EV finds all connectors busy at its arrival in the parking spot, it will be charged during its reserved slot. Vice versa, the occupation becomes very low for both schemes when  $num_{conn} > 10$ , since the EVSE is clearly oversized - in terms of number of connectors - compared to the actual load produced by the EVs. Operational stakeholders interested in infrastructure deployment should then compare the cost of extending the hardware equipment (in terms of additional connectors) against the cost of implementing the reservation and valet service, but this is out of the scope of the current analysis.

## VIII. CONCLUSIONS AND FUTURE WORKS

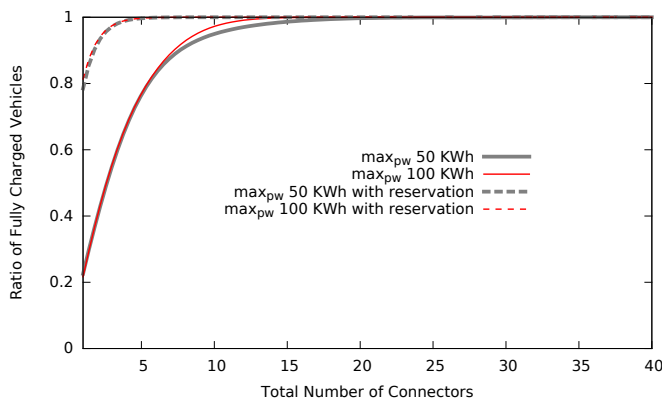
This paper presents a novel simulation framework to analyze the impact and the operations of Electric Vehicles (EVs) in complex, large-scale and heterogeneous environments. The proposed framework integrates models of vehicular mobility, battery charging/discharging and EM-



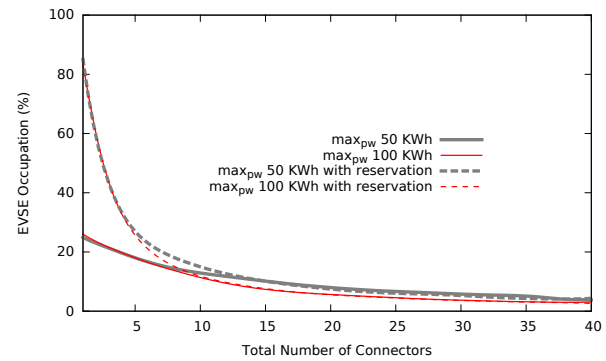
**Figure 17.** Vehicles arrival/departure at the parking spot over hour of the day



**Figure 18.** EVSE load over hour of the day



**Figure 19.** Ratio of fully charged vehicles over total number of connectors ( $num_{conn}$ )



**Figure 20.** EVSE Occupation (%) over total number of connectors ( $num_{conn}$ )

related city services, and it relies on the semantic architecture deployed within the IoE project, through which real and simulated components can be integrated seamlessly. Two use-cases have been discussed: (i) the evaluation of large-scale EM scenarios (i.e. the city of Bologna) through which we derived useful pre-deployment indications about charging infrastructures and reservation services, and (ii) the immersive emulation of EM-related mobile applications through the Mobile Application Zoo (MAZ) concept. Coherently with these two research directions, the future works and open issues embrace on one side the aspects of modeling and simulation of EM scenarios, and on the other the mobile services to assist EVs drivers and also minimize the recharging impact on the grid. Particularly, with reference to the first aspect, we plan to further extend the battery model of the EV by also considering the impact of regenerative braking on energy consumption. At the same time, a major improvement will consist in the further integration of our simulation framework with smart grid simulators and planning tools, in order to analyze the impact of EVs charging operations to the smart grid in terms of reactive power and harmonics propagation. An additional yet more challenging research direction aims to have both the smart grid simulator and our EV simulation platform running simultaneously (and not sequentially, like in Section VII.C), in order to allow the investigation of complex aspects related to energy production and selling, like the energy price dynamics and the utilization of renewable

sources for not only domestic charging services. At present, our framework already provides hooks in the OMNET++ implementation of the EVSE model, to which external software components can be attached in order to read the power request from a charging EV and set the output power. However, there still remains the main issue about the time synchronization, since SUMO/OMNET++ rely on a discrete-event scheduler, while most of the existing smart-grid simulators in the literature (e.g. tools used in [4][5]) are time-driven and based on differential equations, and thus time-stepped synchronization and communication techniques [3] should be implemented. A complementary research direction is focused on the definition of novel and advanced mobile applications for EM scenarios, which can be tested and validated on the proposed framework through the MAZ. To this purpose, we are investigating for example how to extend the battery monitoring application of Section VI.B with battery state prediction algorithms able to suggest charging options on the basis of the user's planned route and of the current SOC.

## IX. ACKNOWLEDGMENTS

This work is supported by the EU ARTEMIS project: "Internet of Energy (IoE) for Electric Mobility" (<http://www.artemisioe.eu>) and by the EIT ICT LABs activity "Planning Tool for EV Deployment and Related User Centric Services" (Activity ID: 14053).

# REFERENCES

- [1] H. T. Zhang, "An overview of smart grid simulator", Proc. of IEEE PESGM, San Diego, USA, 2012.
- [2] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp and L. Mili, "Power system and communication network co-simulation for smart grid applications", Proc. of IEEE ISGT, Anaheim, USA, 2011.
- [3] W. Li and X. Zhang, "Simulation of the smart grid communications: challenges, techniques and future trends", Computers and Electrical Engineering, Article in Press, 2013.
- [4] T. Godfrey, S. Mullen, D. W. Griffith and N. Golmie, "Modeling smart grid applications with co-simulation", Proc. of IEEE SmartGridComm, Gaithersburg, USA, 2010.
- [5] V. Liberatore and A. Al-Hammouri, "Smart grid communication and co-simulation", Proc. of IEEE Energytech, Cleveland, USA, 2011.
- [6] R. Podmore, "The role of simulators for smart grids development", IEEE Transactions on Smart Grids, 1(2), pp. 205-212, 2010.
- [7] K. Mets, T. Verschueren, T. L. Vandoorn and L. Vandevelde, "Integrated simulation of power and communication networks for smart grid applications", Proc. of IEEE CAMAD, Kyoto, Japan, 2011.
- [8] G. A. Pagani and M. Aiello, "Generating realistic dynamic prices and services for the smart grid", JBI technical report, 2013.
- [9] T. Song, D. Kaleshi, R. Zhou, O. Boudeville, J. X. Ma, A. Pelletier and I. Haddadi, "Performance evaluation of integrated smart energy solutions through large-scale simulations", Proc. of IEEE SmartGridComm, Brussels, Belgium, 2011.
- [10] K. Anderson, J. Du, A. Narayan and A. El Gamal, "GridSpice: a distributed simulation platform for the smart grid", Proc. of IEEE MSCPS, Berkeley, USA, 2013.
- [11] J. Hou and X. Guo, "Modeling and simulation of hybrid electric vehicles using HEVSIM and ADVISOR", Proc. of IEEE VPPC, Harbin, China, 2008.
- [12] W. Zhan, M. McDermott, B. Zoghi, and M. Hasan, "Requirement development for electrical vehicles using simulation tools", Proc. of IEEE WCECS, San Francisco, USA, 2009.
- [13] J. P. Trovao, P. G. Pereirinha and H. M. Jorge, "Simulation model and road tests comparative results of a small urban electric vehicle", Proc. of IEEE IECON, Porto, Portugal, 2009.
- [14] S. Onoda, S. M. Lukic, A. Nasiri and A. Emadi, "A PSIM-based modeling tool for conventional, electric and hybrid electric vehicles studies", Proc. of IEEE VTC-Fall, Vancouver, Canada, 2002.
- [15] D. W. Dennis, V. S. Battaglia, and A. Belanger, "Electrochemical modeling of lithium polymer batteries", Journal on Power Sources, 110(2), pp. 310-320, 2002.
- [16] J. Newan, K. E. Thomas, H. Hafezi and D. R. Wheeler, "Modeling of lithium-ion batteries", Journal on Power Sources, 119(131), pp. 838-843, 2003.
- [17] R. C. Kroeze and P. T. Krein, "Electric battery model for use in dynamic electric vehicle simulations", Proc. of IEEE PESC, Rhodes, Greece.
- [18] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban Mobility", International Journal On Advances in Systems and Measurements, 5 (3&4), pp. 128-138, 2012.
- [19] J. R. Pillai and B. Bak-Jensen, "Impacts of electric vehicle loads on power distribution systems", Proc. of IEEE VPPC, Lille, France, 2010.
- [20] M. Levesque, D. Q. Xu, G. Joos and M. Maier, "Co-simulation of PEV coordination schemes over a FiWi smart grid communications infrastructure", Proc. of IECON, Montreal, Canada, 2012.
- [21] S. Karnouskos and T. Nass De Holanda, "Simulation of a smart grid city with software agents", Proc. of IEEE EMS, Athens, Greece, 2009.
- [22] M. Gharbaoui, B. Martini, R. Bruno, L. Valcarengi, M. Conti and P. Castoldi, "Designing and evaluating activity-based electric vehicle charging in urban areas", Proc. of IEEE IEVC, Santa Clara, USA, 2013.
- [23] R. Maia, M. Silva, R. Araujo and U. Nunes, "Electric vehicle simulator for energy consumption studies in electric mobility systems", Proc. of IEEE FISTS, Vienna, 2011.
- [24] A. Hess, F. Malandrino, M. B. Reinhardt, C. Casetti, K. A. Hummel, J. M. Barcelo-Ordinas, "Optimal deployment of charging stations for electric vehicular networks", Proc. of Urbane, Nice, France, 2012.
- [25] A. Schlote, E. Crisostomi, S. Kirkland and R. Shorten, "Traffic modeling framework for electric vehicles", International Journal of Control, 85(7), pp. 880-897, 2012.
- [26] C. Lewandowski, J. Schmutzler and C. Wietfeld, "A simulation environment for electric vehicle charging infrastructures and load coordination", GI Jahrestagung, 10(1), pp. 479-484, 2010.
- [27] Y. C. Yeh, G. G. Meyer, Z. Meyer-Zaho, and M. S. Tsai, "Simulation and evaluation of charging control systems for electric vehicle car parks", Proc. of IEEE PowerTech, Grenoble, France, 2013.
- [28] F. Nemry and M. Brons, "Plug-In Hybrid and Battery Electric Vehicles - Market penetration scenarios of electric drive vehicles", Technical Note from the European Commission Joint Research Centre (JRC) and the Institute for Perspective Technological Studies (IPTS), 2010.
- [29] "Global EV Outlook", Annual Report from the International Energy Agency (IEA), 2013.
- [30] U.S. National Energy Technology Laboratory, "Assessment of Future Vehicle Transportation Options and Their Impact on the Electric Grid Report", DOE/NETL-2010/1466, 2011.
- [31] Internet of Energy (IoE) - ARTEMIS European Project. Project Website: <http://www.artemis-ioe.eu/>.
- [32] A. Varga and R. Hornig. An Overview of the Omnet++ Simulation Environment. Proc. of ICST Simutools, Marseille, France, 2008.
- [33] A. Wegener, M. Piorkowski, M. Raya, H. Hellbruck, S. Fischer and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators", Proc. of ACM CNS, Ottawa, Canada, 2008.
- [34] L. Bedogni, L. Bononi, M. Di Felice, A. D'Elia, R. Mock, F. Montori, F. Morandi, L. Roffia, S. Rondelli, T. Salmon Cinotti, F. Vergari, "An Interoperable Architecture for Mobile Smart Services over the Internet of Energy", Proc. of IEEE SCUCA, Madrid, Spain, 2013.

- [35]Coulomb Technologies, Inc. The ChargePoint Application. <https://www.chargepoint.net.au/>.
- [36]Inc. ECotality. The Blink Mobile Application. Project Website: <http://www.ecotality.com/>.
- [37]The OpenStreetMap Project. Website: <http://www.openstreetmap.org>
- [38]L. Bedogni, A. Trotta, M. Di Felice, L. Bononi, "Machine-to-Machine Communication over TV White Spaces for Smart Metering Applications," in Proc. of IEEE ICCCN, Nassau, Bahamas, 2013.
- [39]C. Sommer, R. German and F. Dressler, "Bi-directionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis", IEEE Transactions on Mobile Computing, 10(1), pp. 3-15, 2011.
- [40]The EnelDrive Application. Project Website: <http://www.eneldrive.it/>.
- [41]ISO 15118 Standard, Road vehicles -- Vehicle to grid communication interface -- Part 1: General information and use-case definition, [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=55365](http://www.iso.org/iso/catalogue_detail.htm?csnumber=55365)
- [42]F. Morandi, L. Roffia, A. D'Elia, F. Vergari, and T. Salmon Cinotti. RedSib: a Smart-M3 semantic information broker implementation. Proc. of FRUCT, Oulu, Finland, 2012.
- [43]J. C. Ferreira, V. Monteiro and J. L. Afonso, Mobile geographic range prediction for electric vehicles, Proc. of IEEE CETC, Lisbon, Portugal, 2011.
- [44]C.-H. Yang, G. Zhavelova, C. -W. Yang and V. Vyatkin, "Cosimulation environment for event-driven distributed controls of smart grid", IEEE Transactions on Industrial Electronics, 9(4), pp. 2124-2136, 2013.
- [45]F. Andren, T. Strasser, S. Rohjans and M. Uslar, "Analyzing the need for a common modeling language for Smart Grid applications", Proc. of IEEE INDIN, Bochum, Germany, 2013.
- [46]PSS SINCAL Platform, Network planning and analysis tool, [http://www.simtec.cc/sites\\_en/sincal.asp](http://www.simtec.cc/sites_en/sincal.asp)
- [47]L. Bedogni, L. Bononi, A. D' Elia, M. Di Felice, S. Rondelli, T. Salmon Cinotti, A Mobile Application to Assist Electric Vehicles' Drivers with Charging Services, Proc. of IEEE NGMAST, Oxford, UK, 2014.
- [48]L. Bedogni, L. Bononi, A. D' Elia, M. Di Felice, M. Di Nicola, T. Salmon Cinotti, Driving Without Anxiety: a Route Planner Service with Range Prediction for the Electric Vehicles, Proc. of IEEE ICCVE, Wien, Austria, 2014.