An Interoperable Architecture for Mobile Smart Services over the Internet of Energy

Luca Bedogni, Luciano Bononi, Marco Di Felice, Alfredo D’Elia, Randolf Mock, Federico Montori, Francesco Morandi, Luca Roffia, Simone Rondelli, Tullio Salmon Cinotti, Fabio Vergari
DISI, University of Bologna, Italy
{lbedogni,bononi,difelice,montori,rondelli}@cs.unibo.it
ARCES, University of Bologna, Italy
{adelia,fmorandi,roffia,tsalmon,fvergari}@arces.unibo.it
Siemens Munich, Germany
{randolf.mock@siemens.com}

October 1, 2013

Abstract

The Internet of Energy (IoE) for Electric Mobility is an European research project that aims at deploying a communication infrastructure to facilitate and support the operations of Electric Vehicles (EVs). In this paper, we present three research contributions of IoE. First, we describe a software architecture to support the deployment of mobile and smart services over an Electric Mobility (EM) scenario. The proposed architecture relies on an ontology-based data representation, on a shared repository of information (Service Information Broker), and on software modules (called Knowledge Processors -KPs) for standardized data access/management. As a result, information sharing among the different stakeholders of the EM scenario (i.e. EVs, EVSEs, City Services, etc) is enabled, and the interoperability of smart services offered by heterogeneous providers is guaranteed by the common ontology. Second, we rely on the proposed architecture to develop a remote charging reservation system, that runs on top of mobile smartphones, and allows drivers to monitor the current state-of-charge of their EV, and to reserve a charging slot at a specific EVSE. Finally, we validate our architecture through a benchmark framework, that supports the embedding of mobile EV applications and of real KPs into a simulated vehicular scenario, including realistic traffic, wireless communication and battery models. Evaluation results confirm the scalability of our architecture, and the ability to support EVs charging operations on a large-scale scenario (i.e. the downtown of Bologna).
1 Introduction

Nowadays, sustainable urban mobility is considered a key component of the Smart Cities initiative. Among the worldwide approaches to improve the energy efficiency of transportation systems while reducing the carbon emissions, Electric Mobility (EM) has gained significant investments from both car manufacturers and governments in these last years [4].

However, despite the initial optimism regarding the potentials of EM, recent market analysis [13][15] have demonstrated that the introduction of EVs might be quite limited in the next decade, unless adequate services will be deployed to facilitate and support the management of EVs by end-users. This is confirmed, among others, by the survey conducted by the U.S. National Energy Technology Lab, according to which 70 percent of people would not purchase an EV due to doubts on the availability of charging stations close to their needs [15].

Several European initiatives and projects have been activated to cope with these limitations [10][11]. At the same time, it is worth to notice that realistic EM scenarios are usually characterized by a multitude of stakeholders (e.g. EV drivers, EVSE providers, power producers, etc) involved on the EM management. Although specific applications have been proposed on small-scale scenarios [5][6][8][13][18], the interoperability among different actors using heterogeneous technologies/devices (e.g. EV models, charging technologies, etc) is far to be guaranteed.

The Internet of Energy (IoE) for Electric Mobility project [1], which is funded by the European Union and comprises 40 partners of 10 European countries, aims to fill this gap, by developing hardware, software and middleware solutions that will provide an interoperable communication infrastructure between the (heterogeneous) stakeholders of a smart-grid scenario. In this paper, we present three contributions of the IoE project regarding the deployment of such infrastructure. First, we propose a software architecture to support the operations of EM services accessible by EV users through mobile smartphones. The architecture foresees the presence of three different actors: i.e. Electric Vehicles (EVs), Electric Vehicle Supply Equipments (EVSEs), and City Services (CSs), connected through an heterogeneous wireless/wired communication network. A power-specific ontology is defined to represent all the information of the smart-grid scenario in a uniform and unambiguous way. Moreover, a City Service Information Broker (CSIB) is introduced to allow information storing and sharing among the different actors based on the SMART-M3 technology [17]. Software agents called Knowledge Processors (KPs) are used by each agent to retrieve/insert/manage data from/into the SIB through a common interface, thus allowing the dynamic definition of new City Services (CSs), as well as the interoperability among services managed by different providers. Second, we describe an on-demand charging reservation system based on the SIB-based architecture. At the EV driver side, a mobile application -running on the driver’s smartphone- autonomously gathers and monitors the state-of-charge of the EV,

\footnote{This behaviour is referred to as "driver’s range anxiety" in [15].}
and interacts with the CSIB to reserve a charging slot on a specific EVSE of the scenario. Leveraging the information stored by EVSEs and EVs, the CSIB can assist the driver in selecting the most suitable charging slot, based on driver’s preference (e.g. minimum waiting time or closest EVSE), as well as on city-whole needs (e.g. traffic load-balancing). Finally, in order to test the SIB-based architecture scalability, we have developed a benchmarking framework, that allows the embedding of Android-based EV applications and of real KPs into a simulated vehicular scenario, characterized by realistic traffic, wireless communication and EV battery models. Using realistic synthetic EM environments, we are able to test the correctness of EV mobile applications, by attaching each client to a simulated EV, and at the same time we can evaluate the scalability of the architecture under varying traffic and network loads produced by charging reservation requests from the EVs.

The rest of the paper is organized as follows. In Section 2, we review the state-of-art on the deployment of mobile services for EM scenarios, focusing on ontology-based approaches. In Section 3, we introduce our software architecture, which is then used to deploy the charging reservation service (described in Section 3.1). Section 4 presents the simulation framework, which is used to emulate EM-related services (Section 5) and to evaluate the IoE architecture scalability (Section 6). Conclusions follow in Section 7.

2 Related works

In these last years, several EM management systems have been proposed to support and facilitate the charging operations by EVs. Most of these systems rely on a service-oriented architecture, and provide a Web-based interface to be always reachable by end-users. Among the worldwide initiatives, we cite the CarStation [5] and the ENELDrive [9] projects, which offer a Web mapping service for EV charging stations. Given the popularity and pervasiveness of mobile devices (e.g. smartphones), some Android and IoS applications have been proposed to help the driver in finding the closest EVSE at his location, and to reserve a charging slot. For instance, the ChargePoint App [6] by Colom-bus Technologies, and the Blink Mobile Application [8] by ECOItality allow to find charging stations, to receive charging status update, and to dynamically route to the chosen station. In [13], these services are enriched with a charging scheduler, that takes into account the energy price curve, and with a user-defined route calendar, through which the application can autonomously deduce the charging needs and trigger charging events. Finally, the authors of [18] analyze different network protocols to manage the communication between the EV and the EVSEs, considering the trade-off between latency time of EVs, and power balancing between the EVSEs. However, all the applications proposed so far are tailored for a specific EV infrastructure provider, and none of them provides facilities to manage and share data coming from different providers, and from different EV models. Conversely, enabling cross-domain information interoperability on EM scenarios constitutes the main goal
of our work. To this aim, the Semantic Web [23] provides an-ontology based extendible framework to integrate independently developed software agents on the basis of a shared knowledge base. Semantic ontologies have been successfully applied to different domains (e.g. health-care, transportation, etc), including the smart-grid one [16]. Moreover, ontology-based middleware solutions have been proposed to support interoperability at the information level among different software agents, abstracting from the details of the communication, and the software/hardware characteristics of the devices. Among others, we cite the Smart-M3 middleware [17], that enables information sharing among heterogeneous devices through a standardized access protocol and lightweight semantic techniques for data self-description. Many applications and demonstrations of the Smart-M3 platform have been developed so far for smart-space scenarios (like biomedical and environmental scenarios [7][22]), characterized by the need to merge sensor data from different producers. In this paper, we rely on Semantic Web ontologies and Smart-M3 technology to support the deployment of smart services over the IoE scenario, as explained in the Section below.

3 Architecture

To manage the electric-mobility scenario, characterized by a high variety of domains, platforms and stakeholders, we decided to use the Smart-M3 interoperability platform as the core of our architecture. According to the Semantic Web paradigm and as specified by the Resource Description Framework (RDF), in Smart-M3, information is represented as a set of triples corresponding to a directed labeled graph where nodes and arcs are univocally identified by URIs. The semantic graph is based on ontologies [12] i.e. machine interpretable domain descriptions, hosted by a Semantic Information Broker(SIB) [14]. The platform features the subscribe-notify mechanism to support application reactivity to context changes. Software agents named Knowledge Processors (KP) are developed according to the separation of concern principle and they are interoperable and extensible to a large extent. Developers are provided with a set of APIs, available for several popular programming languages. The APIs expose RDF sub-graphs with a high level perspective and implement the Smart Space Access Protocol (SSAP), hiding the details of the XML messages transported over TCP. Legacy devices can be easily adapted to the platform by using dedicated KPs to share the legacy context and status, thus a gradual migration to Smart-M3 based frameworks is possible.

The proposed Smart-M3 based architecture for mobile service deployment in the IoE scenario is shown in Figure 1. The City Service Information Broker is an instantiation of the Smart-M3 interoperability component i.e. the SIB. The city services, which need both to be visible and to have visibility of the whole city context, are implemented through KPs performing data mining on the SIB. Relevant vehicle data are collected and sent to the City Service Information Broker by the user smartphone. This may occur with or without the addition of a software component to the EVs on-board telematics unit. The
The first approach is unobtrusive to the EV but implies the smartphone support for the many protocols and data representations exposed by diversified vehicles and car manufacturers. In this case, a driver might for instance download the interface to its EV from an application store. The second approach adds an open EV information broker - i.e. a local SIB - to the EV on-board unit. This powerful but obtrusive solution has already been experimented in the SOFIA project [20]. The smartphone bridges the gap between the EV and the City SIB where all information about the global context is stored. City services are programs with global visibility: they can aggregate raw data and provide complex functionalities as well as more valuable information at a higher level of abstraction. An example of city service for dynamic route planning assistance to EVs is described in Section 3.1. Other entities relevant to the application scenario e.g. Electric Vehicle Supply Equipment (EVSE), can be integrated in the architecture by using adapter Knowledge Processors (KPs) to translate legacy information into RDF sub-graphs. In the proposed architecture, many heterogeneous entities can produce, consume and aggregate information as actors of a machine-to-machine configurable ecosystem. A well-defined shared domain vocabulary facilitates the integration of independently developed modules (KPs). The ontology driven approach combined with the Smart-M3 platform makes the architecture extensible and evolvable. By extending the ontology, new concepts of the EM domain can be represented. At the same time, the implementation of additional aggregator KPs may extend the reasoning capabilities as well as the supported functionalities of the city services and of the EVSE providers. Being the domain description based on machine interpretable concepts, relationships, individuals and statements, relevant context extensions, such as the registration of new users or EVSEs, can be represented with specific sub-graphs which are
dynamically and automatically managed by the system without code modifications.

In the architecture shown in Figure 1, the mutual understanding between all actors is enabled by the domain ontology, designed from the general EM domain concepts and from typical EM-related application requirements. The class taxonomy tree (shown in Figure 2) can be roughly 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, like the ones described in Section 3.1. 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 into individuals and statements according to the ontology, the resulting RDF graph can be then explored by software agents using the SPARQL query language.

![Figure 2: The Ontology class tree for the EM domain.](image)

3.1 Charging Reservation and Route Planning Service

Although the architecture defined so far is general, and can support several mobile services for the EM scenario, in this paper we provided a proof-of-concept of its scalability and correctness by considering a target city service, i.e. the remote reservation system. This system enables remote reservation of charging slots by mobile applications running on EV drivers’ smartphone, and also provides route-planning assistance to select the most suitable EVSE based on drivers’ preferences. To this aim, all architectural components defined in the previous Section are integrated together. The EVSE-adapter KPs publish in the City SIB the information relevant to the application i.e. status, location, charge type, price and current reservation list. The user smartphone application collects vehicle data directly from the EV and notifies the user when the battery
charge reaches a threshold. The notified user can request recharge operation to the city service attaching additional information like current position, battery status, and recharge preferences (e.g., recharge time). The city service analyzes the user request, computes the best available options over the whole city context and provides the processed alternatives. Once the user has received the list of EVSEs, s/he can decide the most suitable one according to different policies (i.e., closest EVSE, EVSE with minimum queuing time, EVSE with lowest energy price) and waits for the final confirmation from the city service. This final confirmation is mandatory in order to avoid synchronization issues due to the concurrency of user requests. The screen-shot of the mobile application is shown in Figure 5. The sequence diagram of the described process is shown in Figure 3: here the interactions with the SIB, including the notification messages used to reduce data traffic and to speed up the service, are enlightened.

Figure 3: The sequence diagram of a SIB request.

4 Validation

Generally speaking, the evaluation of mobile services over large-scale EMs scenarios is highly challenging, since building a real test-bed with a high number of EVs and EVSEs might be impractical. For this reason, we developed an integrated simulation framework for IoE scenarios, through which we evaluated the performance of mobile services built over the IoE architecture described in Section 3. More specifically, the simulation framework allows to model most of the main components of an EM scenario, including the EV charging operations at the EVSEs, the EV mobility (and thus the battery discharge) and the wireless
network communication between the EV and the EVSE. At the current stage of implementation, our framework can be used both for benchmarking and for emulation purposes of EM-related mobile services. In the first use case, the simulation platform is used to generate synthetic environments, characterized by realistic vehicular traffic conditions, EVs charging states, and EVSEs availability, through which we can feed the City SIB with synthetic EM-related data, and thus test the scalability of mobile services under intensive load and traffic conditions. In the second use case, the simulation framework is used to embed EV-related Android applications into the simulated EM environment. As a result, an user can check the status of the (simulated) EV on its smartphone, and at the same time s/he can schedule new simulation events by interacting with the mobile application (e.g. modify the route of a simulated EV by choosing a specific charging station).

Figure 4: The IoE Simulator Architecture.

In Figure 4 the whole IoE simulator platform is shown, highlighting the main components involved. We built an integrated platform composed by SUMO [3] and Omnet++ [21], connected through the VeinS [19] framework. The SUMO tool briefly described in Section 4.2 models the urban traffic mobility, while the Omnet++ tool (Section 4.1) models the operations of the EVSEs, the EVs, and the request/response communication among them as described in Section 3.1. A realistic battery model provided by SIEMENS (and described in Section 4.3) is connected to Omnet++, and manages the discharge events of the EV vehicles and the charge events at the EVSEs. A City SIB (using the SMART-M3 technology and the ontology described in Section 3) is used as a global information storage. The City SIB is dynamically populated with the information coming from the simulated EVSEs (e.g. charging station position and energy price) as they were information coming from real scenario, i.e. the global architecture is fully transparent to the synthetic environment. Moreover, each simulated EV publishes its information about current position and state of charge on a second SIB (called Dash SIB in Figure 4), in order to make it available to the
smartphone application, through the abstraction of Mobile Application Zoo described in Section 5. Needless to say, the presence of the Dash SIB has been introduced for simulation purposes only, and it might not be necessary in a real deployment on an EV, since the mobile smartphone might gather the information needed about its position and charge status directly from the can bus of the vehicle or through its own sensors (e.g. the GPS). In the following Sections, we further present the components of the IoE simulator platform.

4.1 Omnet++
Omnet++ is a discrete-event simulator. We extended it with the models of EVSEs (including the management of the EVs queues) and of the EVs, and we integrated it with the battery charging/discharging models described in Section 4.3. Moreover, we modeled in Omnet++ the network communications between the EVs and the SIB for the charging reservation and route planning services described in Section 3.1.

4.2 SUMO
SUMO is used as vehicles’ mobility simulator. For our analysis, we considered a large-scale scenario (i.e. the downtown of Bologna), with a realistic street map (imported from the OpenStreet Map project 2) and realistic traffic density data, gathered through induction loops placed at different locations of the city. However, we highlight that our simulation platform can easily be adapted to work on any vehicular scenario, by changing the street map and the positions of the EVSEs.

4.3 Battery model
The battery model governs the battery discharge operations of EVs, and the charging actions at the EVSEs. It is developed and provided by SIEMENS as an external library to facilitate the model updates. The discharging model computes the current EV charge state taking into account the mobility parameters (e.g. speed, acceleration), the vehicle’s characteristics (e.g. weight) and battery physical properties (e.g. temperature). Similarly, the charging model computes the time needed to fully or partially re-charge an EV, based on EVSE characteristics (like min-max capacities).

5 The Mobile Application Zoo for EM
The Mobile Application Zoo is built as a sandbox through which Android-based applications can be embedded into a simulated EM environment. As a result, developers can test their mobile applications over realistic and large-scale EM scenarios, in a seamless way and with minimal changes required with respect to

2http://www.openstreetmap.org
the deployment on a real scenario [2]. This is made possible by the adoption of a common data ontology between all the components of the framework (including both real and simulated components), and by the utilization of KP libraries for mobile applications, so that the smartphone can communicate directly with the Dash SIB, and retrieve the information produced by the simulation. No direct communication is involved between the simulation tools (Omnet++/SUMO) and the smartphone. At periodic intervals, each simulated EV inserts into the Dash SIB the information about its own location and its state of charge. This information are then retrieved by the mobile application, and shown to the end-user as shown in Figure 5a. Moreover, the mobile application can produce and insert new information into the Dash SIB during the simulation time. Leveraging the bidirectionality of the communication between the Dash SIB and the mobile application, new simulation events can be scheduled at runtime, like re-route events determined when the user chooses a charging station (Figure 5b). In this case, the user’s action produces the insertion of a new tuple into the Dash SIB relative to a charging request, that is then read by the Omnet++ module to re-route the simulated EV vehicle to the chosen EVSE.

6 Performance Evaluation

To characterize the performance of our service architecture we first introduce the proposed evaluation metrics, then we describe the results of performance tests and we discuss the architecture scalability comparing them with the real
scenarios requirements. Performance is based on the execution time of the SIB interaction primitives (i.e. queries, updates, subscribes, notifies) and also on the impact of the type of connectivity and communication technology used. The service evaluated is the charging reservation system described in Section 3.1. This service permanently subscribes to the SIB and is triggered by notifications. A service, in fact, needs to be notified to react when a new service request occurs (Figure 3) and when relevant context information changes. The client adds two not concurrent subscriptions respectively to receive the available options and the final confirmation after the user selection. Consequently, the total number of active subscriptions depends on the amount of services available and on the number of clients concurrently waiting for a response from the service; the latter is related to the requests distribution, and it is negligible as long as the service handling time is short compared to the interval between distinct requests. In every Smart-M3 based service, the most frequently used primitives are inserts (e.g. of reservation requests and options), removes (to keep only updated and relevant information in the SIB), queries and subscriptions. The subscription handling algorithm adds an overhead to every insert and remove because all inserted or removed triples need to be checked by the SIB to see if any notification has to be issued [8]. Insert and remove times are dominant in typical applications, they are similar and both depend on the number of triples involved and on the number of currently active subscriptions. Figure 6 enlightens the performance of the proposed architecture implementation by reporting the insert time as a function of the number of triples for various amounts of active subscriptions. The tests have been performed with a wireless connection between client and SIB, to take into account the network overhead, that cannot be neglected in a mobile scenario.

![Figure 6: The Insert time on the CSIB as a function of the active subscribers.](image)

The EVSE reservation service adds two permanent subscriptions to the total number of subscriptions which is roughly proportional to the number of active services. Assuming the EVSE reservation service to be the only service available, and considering the amount of triples exchanged according to the application
ontology, the total SIB KP communication time to solve a client reservation request is less than 30 ms. Considering the city of Bologna as target scenario, the number of vehicles is approximately 200,000. Assuming a 100% EV penetration, with one charging request per vehicle per day over a time period of 8 hours per day, we would have an average value of less than 8 requests per second, leaving the SIB access channel free for approximately 75% of the time, therefore all requests could be managed and some concurrent additional service could also be supported. We can then conclude that the proposed architecture implementation may support EV charging services over variable scale cities. However, increasing the recharging demand or the number of available services may require further work to partition the workload among multiple SIBs.

7 Conclusions

In this paper, we have presented three research contributions of the IoE project pertaining to the support and management of mobile smart services for EM scenarios. First, we have introduced an architecture to support data sharing and interoperability among different stakeholders of an EM scenario, based on Semantic Web ontologies and the Smart-M3 technology. Then, we have developed a service to allow drivers to monitor the current state-of-charge of their EV, and to reserve a charging slot at a specific EVSE. Finally, we have evaluated the proposed system through an integrated simulation platform, that allows to embed EM-related Android applications into a synthetic urban scenarios, and to test the services under realistic traffic and network loads. We are currently extending the simulation framework by considering the altitude factor in both the mobility model (that is actually on 2D), and in the discharging model, and we are studying load-balancing techniques to partition the workload among multiple City SIB on large-scale scenarios. Moreover, we are also planning to test our charging reservation application on a real EV testbed.

Acknowledgments

This work is supported by the ARTEMIS project: Internet of Energy (IoE) for Electric Mobility.

References

[1] Internet of Energy (IoE) - ARTEMIS European Project. Project Website: http://www.artemis-joie.eu/


Source: http://www.comuni-italiani.it/037/statistiche/veicoli.html


[20] Smart Objects For Intelligent Applications (SOFIA)- ARTEMIS European project. Project Website: http://www.sofia-project.eu/

