Internet of Things

Franco Cicirelli · Antonio Guerrieri
Carlo Mastroianni
Giandomenico Spezzano · Andrea Vinci
Editors

The Internet of Things for Smart Urban Ecosystems

Springer

Internet of Things

Technology, Communications and Computing

More information about this series at http://www.springer.com/series/11636

Franco Cicirelli · Antonio Guerrieri
Carlo Mastroianni · Giandomenico Spezzano
Andrea Vinci
Editors

The Internet of Things for Smart Urban Ecosystems

Springer

Editors
Franco Cicirelli
Institute for High Performance Computing and Networking (ICAR)
National Research Council (CNR)
Rende (CS), Italy

Antonio Guerrieri
Institute for High Performance Computing and Networking (ICAR)
National Research Council (CNR)
Rende (CS), Italy

Carlo Mastroianni
Institute for High Performance Computing and Networking (ICAR)
National Research Council (CNR)
Rende (CS), Italy

Giandomenico Spezzano
Institute for High Performance Computing and Networking (ICAR)
National Research Council (CNR)
Rende (CS), Italy

Andrea Vinci
Institute for High Performance Computing and Networking (ICAR)
National Research Council (CNR)
Rende (CS), Italy

ISSN 2199-1073 ISSN 2199-1081 (electronic)
Internet of Things
ISBN 978-3-319-96549-9 ISBN 978-3-319-96550-5 (eBook)
https://doi.org/10.1007/978-3-319-96550-5

Library of Congress Control Number: 2018948620

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Nowadays, the exploitation of ICT technologies in urban environments is enabling the realization of smart infrastructures that provide enhanced digital services improving the quality-of-life of citizens and the cities' efficiency.

In this field, significant smart applications include but are not limited to smart parking, traffic management, smart lighting, structural health monitoring, air quality and pollution monitoring, health monitoring, waste management, smart grids and city energy optimization, smart urban drainage networks, building automation, and emergency detection and management.

Since all these applications share a common urban environment and common goals, they need to interact with each other and with citizens, creating a Smart Urban Ecosystem (SUE).

An SUE is a people-centric system of systems needing a tight integration between cyber and physical components for sensing, reasoning, and controlling the urban environment. In this context, The Internet of Things (IoT) constitutes an enabling technology as it bridges the gap between physical things and software components and empowers the cooperation among distributed, pervasive, and heterogeneous entities.

The development of an SUE introduces several challenges, which include heterogeneous system integration, interoperability among different technologies, fault tolerance, scalability, system maintenance, geographical and functional extensibility, social networking, mobile computing, context-aware applications and services, human-in-the-loop modeling and simulation, big data analysis, cloud- and edge-based IoT frameworks and environments, field experiments and testbeds. Moreover, the realization of an SUE requires cross-domain knowledge expertise spanning from computer science to electronic, electrical, civil, hydraulic, and energy engineering.

The main objective of this book is to provide a multidisciplinary overview of methodological approaches, architectures, platforms, algorithms, and applications for the realization of Smart Urban Ecosystems.

The book includes 15 chapters covering three main topics: (i) software architectures and platforms for SUEs, (ii) development approaches and algorithms for SUEs implementation, and (iii) applications and case studies related to specific smart infrastructures and smart cities. A short introduction to the chapters is provided below.

The Chapter "A Social and Pervasive IoT Platform for Developing Smart Environments", by Orazio Briante, Franco Cicirelli, Antonio Guerrieri, Antonio Iera, Alessandro Mercuri, Giuseppe Ruggeri, Giandomenico Spezzano, and Andrea Vinci, gives an overview of the iSapiens platform, which is a Java-based platform specifically designed for the development and implementation of Smart Environments (SEs). iSapiens exploits the Social Internet of Things paradigm that allows to dynamically discover "things" without requiring intervention from humans. iSapiens provides tools for the realization of pervasive SEs and relies on the edge computing paradigm. Moreover, the chapter reviews some SE applications built by leveraging iSapiens features.

The Chapter "Smart City Platform Specification: A Modular Approach to Achieve Interoperability in Smart Cities", by Arianna Brutti, Piero De Sabbata, Angelo Frascella, Nicola Gessa, Raffaele Ianniello, Cristiano Novelli, Stefano Pizzuti, Giovanni Ponti, proposes a development methodology and a modular and scalable multilayered ICT platform to address the problem of cross-domain interoperability in the context of smart city applications. As a distinct feature, the chapter tackles issues about interoperability on the Information and Semantic levels and provides an approach for finding the correct balance between prescriptive and elastic specifications.

The Chapter "Integrated Cyber Physical Assessment and Response for Improved Resiliency", by P. Sivils, C. Rieger, K. Amarasinghe, and M. Manic, provides a summary and analysis of crucial concepts and challenges in assessing cyber-physical degradation, heterogeneous data fusion, and visualization under a smart city IoT architecture. The main objective is to provide a basis for enhancing the effectiveness of human response to physical and cyber events.

The Chapter "On the Integration of Information Centric Networking and Fog Computing for Smart Home Services", by Marica Amadeo, Andrea Giordano, Carlo Mastroianni and Antonella Molinaro, focuses on the role that can be assumed by the Information Centric Networking (ICN) paradigm to support future Internet communications and data delivery in smart urban ecosystems, including smart home/building services. The integration of ICN with Cloud/Fog resources is also discussed in the chapter, and a reference architecture is presented as a proof-of-concept, together with a preliminary testbed.

The Chapter "Optimal Placement of Security Resources for the Internet of Things", by Antonino Rullo, Edoardo Serra, Elisa Bertino, and Jorge Lobo, deals with the problem of efficiently and effectively securing IoT networks by carefully allocating security resources in the networked area. The problem is modeled according to the game theory, and a Pareto-optimal solution is provided, in which the cost of the security infrastructure and the probability of a successful attack are minimized. In the chapter, authors make a distinction between static and mobile networks, and address the problem of computing the best allocation plan with different approaches.

The Chapter "Embedding Internet-of-Things in Large-Scale Socio-technical Systems: A Community-Oriented Design in Future Smart Grids", by Yilin Huang, Giacomo Poderi, Sanja Šćepanović, Hanna Hasselqvist, Martijn Warnier, and Frances Brazier, focuses on the design of large-scale Socio-technical Systems (STS) relying on Internet of Things technologies. The design of such systems, especially in a complex social context like that of Urban Ecosystems, requires the use of suitable methodological approaches that have not yet entered into the mainstream of design practice. The chapter reviews the literature and presents some lessons learned in adopting an STS-based approach for the design of a community-oriented smart grid application.

The Chapter "Aggregation Techniques for the Internet of Things: An Overview", by Barbara Guidi and Laura Ricci, starts from the consideration that an Internet of Things environment can connect a very large number of sensors, so generating a huge amount of data. Aggregation techniques are required to reduce the size of data to be transmitted and stored, while maintaining a reasonable level of approximation. The chapter offers an overview of a set of aggregation techniques that can be exploited in the IoT, ranging from Space-Filling Curves to Q-digest, Wavelets, Gossip aggregation, and Compressive Sensing.

The Chapter "Swarm Intelligence and IoT-Based Smart Cities: A Review", by Ouarda Zedadra, Antonio Guerrieri, Nicolas Jouandeau, Giandomenico Spezzano, Hamid Seridi, and Giancarlo Fortino, reviews swarm intelligence-based algorithms and related smart city solutions. A swarm intelligence-based framework for smart cities is presented that uses a decentralized control over its components in order to build scalable and flexible smart cities. In addition, a set of trends on how to use swarm intelligence in smart cities, in order to make them flexible and scalable, is investigated.

The Chapter "Cost Saving and Ancillary Service Provisioning in Green Mobile Networks", by Muhammad Ali, Michela Meo, and Daniela Renga, discusses how mobile network operators are required to face huge operational costs, due to the staggering increase in mobile traffic and substantial bandwidth reliability requirements in Smart Urban Ecosystems. The chapter analyzes, for a real scenario, the notable benefits that can be achieved by combining the WiFi offloading approach, the techniques for dynamic adaptation load in a Demand Response context, and the usage of renewable energy sources.

The Chapter "Structural Health Monitoring (SHM)", by Raffaele Zinno, Serena Artese, Gabriele Clausi, Floriana Magarò, Sebastiano Meduri, Angela Miceli, and Assunta Venneri, aims at fitting the structural health monitoring into the Internet of Things. For this purpose, the chapter extensively details, through the description of a real application scenario, all the phases required for developing an effective online and real-time assessment of the structural health of a building.

The Chapter "A Smart Air-Conditioning Plant for Efficient Energy Buildings", by Roberto Bruno, Natale Arcuri, and Giorgio Cuconati, focuses on the correct management of energy fluxes in the context of prosumer systems. The goal is to make effective the process of producing, storing and consuming energy, and increasing users' remuneration. The chapter describes a smart air conditioning system and the correspondent control strategies adopted for its management. The system is based on the employment of photovoltaic-driven heat pumps with thermal storage connected to a radiant emission system.

The Chapter "A Comprehensive Approach to Stormwater Management Problems in the Next Generation Drainage Networks", by Patrizia Piro, Michele Turco, Stefania Anna Palermo, Francesca Principato and Giuseppe Brunetti, shows how the next generation of urban drainage networks can benefit from Internet of Things and ICT technologies. The chapter describes two innovative approaches for managing drainage networks, exploiting a decentralized real-time control system and low-impact development techniques.

The Chapter "Cooperative Video-Surveillance Framework in *Internet of Things* (IoT) Domain", by A. F. Santamaria, P. Raimondo, N. Palmieri, M. Tropea, and F. De Rango, presents the main issues related to the design of an architecture for a smart cooperative video-surveillance system. The chapter shows a surveillance system based on a cooperative tracking among cameras and involves advanced techniques of detection and tracking. In addition, the chapter presents a significant use case showing how an anomaly can be detected, followed by a set of cameras, and managed by generating alerts and messages without human intervention.

The Chapter "Personal Connected Devices for Healthcare", by Adina Riposan-Taylor and Ian J. Taylor, offers a comprehensive survey on personal connected health technologies, which are fast becoming integral in a person's daily life to help improve their health and well-being. The chapter provides a broad overview of a range of devices that allow people to monitor their own health and/or provide self-therapeutic benefits using biofeedback or neurofeedback to control their own physiologic functions. Specific focus is given to the techniques adopted to provide access to data, to the software used for data integration, to connectivity issues, and to user interfaces.

The Chapter "Evacuation and Smart Exit Sign System", by V. Ferraro and J. Settino, exploits the Internet of Things for the realization of emergency evacuation systems. The chapter proposes a smart system that relies on a set of sensors and smart exit signs which are coordinated by a reliable and dynamic evacuation algorithm, which is capable of fast adapting to changing conditions during the evolution of an emergency.

We would like to thank all the book contributors, the anonymous reviewers, and Prasanna Kumar Narayanasamy from Springer for his precious support during the publication process.

Rende (CS), Italy

Franco Cicirelli
Antonio Guerrieri
Carlo Mastroianni
Giandomenico Spezzano
Andrea Vinci

Contents

A Social and Pervasive IoT Platform for Developing Smart Environments

Orazio Briante, Franco Cicirelli, Antonio Guerrieri, Antonio Iera, Alessandro Mercuri, Giuseppe Ruggeri, Giandomenico Spezzano and Andrea Vinci

Abstract Nowadays, the increasing in the use of Internet of Things (IoT) devices is growing the realization of pervasive Smart Environments (SEs) and Smart Urban Ecosystems, where all the data gathered by the "Things" can be elaborated and used to improve the livability, the safety and the security of the environment, and to make inhabitants lives easier. Many efforts have been already done in the direction of SEs development and in the implementation of platforms specifically designed for SE realization. Anyway, such efforts miss of solutions regarding the interoperability among the realized SEs and other third-part "Things". This chapter gives an overview of iSapiens, which is a Java-based platform specifically designed for the development and implementation of SEs. iSapiens tries to overcome the interoperability issue by leveraging the Social Internet of Things (SIoT) paradigm that allows to dynamically include in an SE the new "Things" that can appear in an environment without requiring interventions from humans. iSapiens provides tools for the

F. Cicirelli · A. Guerrieri (✉) · A. Mercuri · G. Spezzano · A. Vinci
CNR - National Research Council of Italy - Institute for High Performance Computing and Networking (ICAR), Rende (CS), Italy
e-mail: guerrieri@icar.cnr.it

F. Cicirelli
e-mail: cicirelli@icar.cnr.it

A. Mercuri
e-mail: mercuri@icar.cnr.it

G. Spezzano
e-mail: spezzano@icar.cnr.it

A. Vinci
e-mail: vinci@icar.cnr.it

O. Briante · A. Iera · G. Ruggeri
University Mediterranea of Reggio Calabria - DIIES Department, Reggio Calabria, Italy
e-mail: orazio.briante@unirc.it

A. Iera
e-mail: antonio.iera@unirc.it

G. Ruggeri
e-mail: giuseppe.ruggeri@unirc.it

F. Cicirelli et al. (eds.), *The Internet of Things for Smart Urban Ecosystems*,
Internet of Things, https://doi.org/10.1007/978-3-319-96550-5_1

realization of pervasive SEs and relies on the edge computing paradigm. Such paradigm is extremely important in a distributed system since it allows to use distributed storage and computation at the edge of a network, so reducing latencies with respect to move all the executions and storages in the cloud. Moreover, the chapter will review some SE applications realized by exploiting iSapiens concepts.

Keywords Smart environments · Social internet of things
Edge computing · Multi agent systems · IoT development platforms · IoT-based applications

1 Introduction

Smart Environments (SEs) are pervasive and highly dynamic systems involving a set (possibly huge) of devices which can interact each other, and which can be dynamically added to and removed from the system itself. SEs usually exploit the IoT paradigm for orchestrating heterogeneous devices [1, 2]. SEs final aim is to furnish advanced cyber-physical services to their inhabitants and administrators. Besides being dynamical in terms of physical things, such systems are required to be also dynamic regarding their functionalities, that should evolve according to changes in the SEs finalities. As a consequence, an SE system should have the capability of adding, updating, and removing functionalities depending on the goals and on the updates in the available devices. Among SEs, Smart Urban Ecosystems are gaining importance as a new research topic, involving the same issues presented for the SEs, but typically on a larger geographical scale, involving a wider number of users, and having a greater impact on people lifestyle.

Moreover, Smart Urban Ecosystem development requires to address issues related to dynamic and open scenarios, such as entity discovery, trustworthiness, interoperability, data processing and management. Furthermore, it is of paramount importance to have methodological guidelines and tools to foster the development of such systems by dealing with their complexity [3, 4].

This chapter is devoted to the review of iSapiens [5], which is an agent-based platform taking into account the above mentioned issues. This platform addresses data management and processing by coupling the cloud computing paradigm [6] with the edge computing one [7, 8], leveraging the agent metaphor [9], thus providing systems with both high computational power, exploitable for high-demanding tasks, and distributed elaboration capabilities supporting reactive and location-dependent processing. Entity discovery, trustworthiness, and interoperability are realized by integrating the Social Internet of Things (SIoT) computing paradigm [4, 10] with the iSapiens platform. Finally, device heterogeneity is reached by the introduction of specifically designed abstractions, namely Virtual Objects.

Besides providing details about the platform, the chapter also reviews a set of applications and case studies which involve the exploitation of the concepts, the features and the capabilities that are at the basis of the iSapiens platform.

The chapter is organized as follows. Section 2 introduces some related work, Sect. 3 provides details about the whole iSapiens platform, comprehending its architecture, abstractions and SIOT component. Section 4 reviews a set of application built on top of the iSapiens concepts. Finally, some conclusions are given to the reader.

2 Related Work

Smart Environments, IoT-based approaches, and the recent Social IoT para-digm are all based on the pervasive presence of a possibly huge number of heterogeneous smart objects, which are capable to interact and cooperate each other so as to create novel application or services for the user, and reach shared goals. In this direction, the research and industry communities are putting strong efforts, in particular for both the creation of IoT-based platforms for SEs [11–14], and the development and integration of social capabilities in the IoT [15, 16], which can help in the discovery/use of new devices that can enter, move inside, or leave an SE. Furthermore, such technologies and efforts have given a boost in the development of Smart City and, in general, Smart Urban Ecosystems applications.

2.1 IoT and Smart Environments Platforms

Nowadays, several SEs [17] and platforms/middlewares for their creation [18–21] have been introduced with the aim of tackling specific issues of these complex systems. Most of these platforms lack in interoperability and scalability which are important for the realization of comprehensive Smart Urban Ecosystems which use several enabling technologies such as cloud [6] and/or edge computing, and social capabilities [10, 22].

Actually, literature presents many approaches for implementing SEs. A selection of them is presented in the following. In [23] interactions among several bluetooth mobile devices are at the basis of a framework designed for implementing SEs capable of dynamically discover and exploit available bluetooth devices.

In [24], authors focused on system extensibility by introducing a framework which has been created for the support in the creation of context-aware applications.

The authors of [25] introduced a distributed middleware, based on the concept of *active spaces*, aimed at managing resources placed in physical environments. Active spaces are programmable environments where the involved SE devices are connected.

The development of context-sensing and context-aware applications is the aim of the work presented in [26], where they defined a framework providing a set of abstractions suitable for modeling different kinds of SEs.

A middleware for context-aware smart homes has been presented in [19], with the aim of foster the development of smart homes based on the sharing of contextual information among the smart home entities.

The well known *Syndesi* framework [18] has been developed for gathering data to profile users and make available to them customized services which exploit wireless sensor networks for both sensing and actuation purposes.

2.2 Smart Cities and Smart Urban Ecosystems

Regarding smart cities and smart urban ecosystems, which are hot topics for the research community, several works have been presented so far and several platforms implemented [27].

In [28] authors have presented an open source IoT platform for the pervasive use of actuators and sensors in smart cities. Such platform, called *Sentilo*, exploits both Big Data tools and cloud computing to gather, save, and analyze data coming from distributed sensors.

The work in [29] shows an IoT middleware that was developed for the EPIC (European Platform for Intelligent Cities) project. Issues such as heterogeneity, interoperability, extensibility, and (re)configurability were at the basis of the design and development of the middleware.

A platform for the realization of smart city is proposed in [30]. Such platform relies on Big Data analysis for achieving extensibility. Its architecture presents three layers that have been designed for (i) collecting and analyzing information, (ii) aggregating data in order to infer some knowledge, (iii) offering to the users instruments to take computed data.

Authors of [31] have introduced Smart Connected Communities in order to extend the concept of smart city considering also livability, preservation, revitalization, and sustainability of the areas in the city to improve. Smart Connected Community is thought as comprehending not only a specific city area but all its neighborhood. Smart Connected Community is based on a multi-layer architecture in which is specifically taken in consideration the importance of exploiting social capabilities.

2.3 The Social IoT

Recently, the Social Internet of Things (SIoT) paradigm risen to the attention of the the IoT research community by fostering the convergence between typical technologies and solutions of the Social Networks domain and the IoT [10]. The idea behind this paradigm is as simple as tremendously powerful: similarly to what happens in the human communities, the IoT objects are enhanced so to become Social IoT objects capable of establishing mutual social relationships in an autonomous way according to rules set by their owner. The insurgence of such a social network [32] has proven to facilitate resource visibility, service discovery, object reputation assessment, source crowding, and service composition in the IoT [10]. Let's consider as an example, the case of an object dealing with the likely overwhelming task of seeking an information provided by the multitude of its peers, might use its social connection to drive the research process by limiting the scope only to those nodes with mutual social relations, exactly in the same way as humans search over their Social Network platforms, then the complexity and the time duration of the search could be drastically reduced [10]. Also, social objects might asses their reciprocal trustworthiness and reputation based on their mutual social relationships just as human beings mostly do [33–36]. It has been proven that such an approach allows isolating almost any malicious device in a network [37].

So far, five types of relationships have been defined for this new paradigm [10]:

1. *Ownership Object Relationship* (*OOR*): that is created among objects belonging to the same user;
2. *Parental Objects Relationship* (*POR*): that is created among objects produced by the same manufacturer and belonging to the same production batch;
3. *Co-Work Objects Relationship* (*C-WOR*): that is created among objects cooperating towards the provision of the same IoT application;
4. *Co-Location Objects Relationship* (*C-LOR*): that bounds objects that are always used in the same place;
5. *Social Object Relationship* (*SOR*): that connects objects that come into contact, either sporadically or continuously, because their owners come in touch each other.

It is also possible to go beyond the types of relationships so far defined by specifically define special groups to include devices bounded by some common features or finalities. Following the interest the SIoT paradigm has attracted, an open source platform called *SIoT Platform*[1] was developed.

3 iSapiens: A Platform for Smart Environments

The iSapiens platform has been developed with the aim of providing a useful tool to developers and researcher for developing and implementing distributed cyber-physical systems [38–40] and smart environments [20, 41–44], even at large scale, thus enabling the creation of Smart Urban Ecosystems.

iSapiens is an IoT [15, 45] enabling, agent-based platform which exploits the edge computing paradigm [7] for its main operations, and permits the exploitation of the cloud for operations that are high-demanding in terms of storage, memory and computation requirement or that cannot be affordably executed in a distributed environment. The platform tackles with the heterogeneity of physical devices and

[1] Social Internet of Things (SIoT), https://www.social-iot.org.

protocols, which is a common issue of IoT application, by providing specific abstractions, namely Virtual Objects, so as to permit to focus on the functionalities/services that need to be realized in order to enhance the behavior of a specific environment.

Agents are used to implement the functionalities of the system under development. Agents, which suits well with the distributed and pervasive nature of smart environments, are deployed and run on a set of networked computational nodes, located close to the physical devices they have to manage and control. The network of computational nodes implements the edge computing paradigm and enables real-time analysis and a fast sensing-decision-actuation cycle. In addition, agents can also exploit out-of-the-edge services, purposely developed or third-party provided. Such out-of-the-edge services can be cloud-based and include predictive analysis, machine learning, data mining and other services that can take advantage of features and benefits of the cloud. Interoperability issues are dealt with by specific components devoted to integrating agents and virtual objects with the Social Internet of Things.

Figure 1 depicts the three layer of the iSapiens architecture, which are: (i) the physical device layer, (ii) the in-network computing layer, and (iii) the off-network computing layer.

The physical devices layer is populated by physical objects. As stated before, such objects can be heterogeneous in terms of communication protocols and computational capabilities, and span from simple sensors/actuators to more complex smart objects. A physical object offers a one or more physical functionalities, which regard on how it interacts with the environment, and can expose a *Social Senser*, that enables the object to integrate and exploit the social Internet of Things.

The In-network computation layer is composed of a set of computing nodes spread into the environment, each running an iSapiens server consisting of a Virtual Object Container and an Agent Server. The Virtual Object Container is devoted to the management of Virtual Objects, which abstract the physical object in order to hide devices heterogeneity. The functionalities offered by the Virtual Objects are exploited by the agents running on the Agent Server for doing the application business. Agents can directly interact with the Virtual Objects which are co-located in the same iSapiens server. A computational node hosting an iSapiens server is called *iSapiens node*. Social Object Containers, hosted by an iSapiens node or other computational nodes, provide functionalities enabling the integration of physical objects with the SIoT. In particular, the Social Object Container manages Social Objects, which are the social counterpart of the Virtual Objects.

The Off-network computation layer hosts cloud/internet services, which can provide storage, computation or specific information, e.g., weather forecasting. This layer also hosts the SIoT platform, which is the core of the social infrastructure. Through such component, iSapiens agents and iSapiens physical objects can interoperate with entities belonging to other platforms, provided that these entities are SIoT compliant.

All the iSapiens components, i.e., devices, VOs, agents or computing nodes, can be dynamically removed from, added into or updated to a system at runtime.

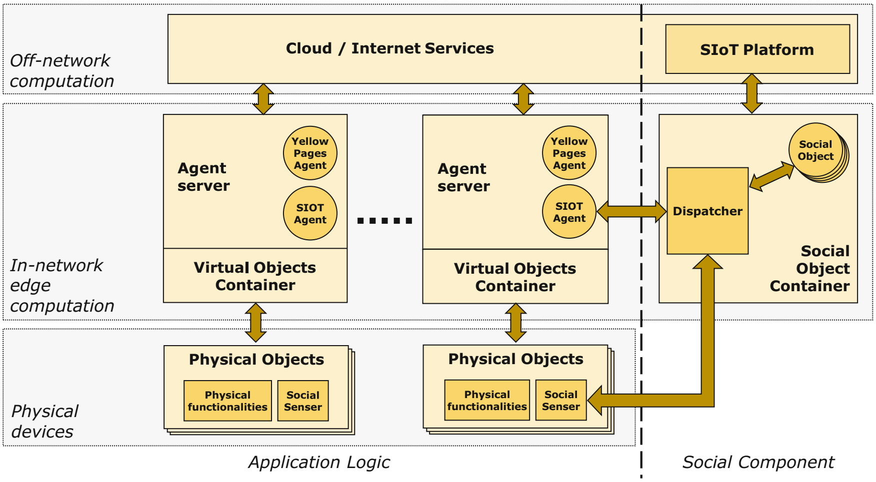

Fig. 1 The *iSapiens* platform and its computational layers

Table 1 The iSapiens platform: benefits and features

Features / Benefits	In-network/Edge Computing	Agents	Virtual Objects	Dynamic Deployment	Yellow Pages / Acquaintances	Off-network/Cloud Computing	SIoT Integration
Functional Extendibility	+	++	++	++	++	+	+
Geographycal Extendibility	+	++	++	++	++	+	+
Online Analytics Support	++	/	/	/	/	+	/
Offline Analytics Support	+	/	/	/	/	++	/
Interoperability	/	/	/	/	/	/	++
Fault Tolerance	++	/	/	/	/	+	+
Heterogeneity Management	/	/	++	/	/	/	+
Software Maintenance	/	++	/	++	++	/	/
Hardware Maintenance	/	/	++	++	/	/	+

This important feature fosters application scalability and extendibility. The current implementation of iSapiens relies on the Java technology.

Table 1 highlights the main benefits provided by iSapiens for the development of smart systems, and relates them with the features offered by the platform.

In the following, we further describe the main entities and mechanisms of the iSapiens platform, that are VOs, Agents, dynamic deployment and SIoT integration.

3.1 Virtual Object Features

Virtual Objects (VOs) have been appositely created in order to offer an instrument for the *devices heterogeneity* management. For this purpose, VOs allow agents to interface with hardware transparently, through APIs provided as common interfaces. Any agent, thanks to VOs, can be connected to a particular device without caring of specific drivers or tackling specific technical issues. VO aim is that of decoupling (both for sensing and actuation) the specific devices from the functionalities offered by such devices. By using VOs, an update or a change of the devices or of the communication protocol used, will concern only the VOs themselves. The *Virtual Object Container* is the entity devoted to the VOs management. It allows to dynamically deploy new VOs and permits them to be addressed by the agents from the agent layer. Asynchronous and synchronous reading of data is supported by VOs. In particular, the publish/subscribe pattern is at the basis of the asynchronous one. Summarizing, VOs expose the features of the physical objects deployed in the environment. Such features are exported into iSapiens and named as *virtual object functionalities* and are connected to actuation and sensing. As an example, through a VO, a smart desk can expose a functionality called *isUsed* that allows agents to understand if someone is sit at the desk.

3.2 Agent Services and Features

An *Agent* [46] is defined ad an entity which autonomously is executed on an Agent Server and which runs its tasks also interacting with other agents that can be both local or remote. An agent residing on an iSapiens Agent Server can use all the functionalities that are exposed by the VOs on the same server. The Agent Server is devoted to support the life cycle of all the agents running on the server itself. Moreover, it permits, also at runtime, the creation of agents.

Agents can communicate through asynchronous messages, that can be both *untimed* or *timed*. These last timed messages hold an information on the time that specifies the time (absolute) in which the information must be given to the recipient agent. Direct acquaintance relationship can be set between agents by using *acquaintance messages*. An acquaintance message can provide information on the role and the identity of a specific agent. These messages are important since they allow to dynamically configure/reconfigure the relations among agents when a fault occurs or if new agents are introduced in the system or if agents are deleted. In order to dynamically discover agents, iSapiens allows the use of a distributed service of *yellow pages* which favors relationships establishment. Each agent can accede to yellow pages and register itself by entering its roles and its properties (i.e., a set of couples formed as <*key,value*>). Once an agent is registered, it can be found/discovered by other agents through the yellow pages (which can be queried).

3.3 SIoT Features

iSapiens implements the Social Internet of Things paradigm by leveraging the features offered by the SIoT Platform that is a cloud-based system that allows objects to autonomously establish social relationships [47]. Social relations between objects are based either on intrinsic properties of the objects such as their location, their producer and their owner, or on number and duration of mutual contacts. Like most human social networking platforms, the SIoT Platform offers the opportunity to define special groups including devices bounded by some common features or finalities. Its main features include: (i) the "Relationship management" to start, update, and terminate relationships among objects, (ii) the "Service discovery" that is finalized to find which objects can provide the required service, (iii) the "Service composition" that enables the interaction among objects and, (iv) the "Trustworthiness management" to understand how the information provided by other members has to be processed. Whenever an object is associated to iSapiens, it is also registered to the SIoT platform and its social profile is created.

The devices connected to an iSapiens node are abstracted as the union of *Virtual Objects* (VOs) (see Sect. 3.1) and *Social Objects* (SOs). All the SOs residing in a specific node are managed by a *Social Object Container*. The SO is responsible to manage all the interactions with the SIoT Platform on behalf of its real counterpart. More in detail, a SO is responsible to constantly update the object profile on SIoT by periodically updating all the relevant information about the object it represents. To carry out its tasks, a SO exploits the information provided by the *Social Senser*, which is installed on the real device and periodically overhears the transmission over the Wi-Fi and Bluetooth interfaces to search for other nearby social-enabled devices. This information, once forwarded by the social object to the SIoT Platform, is used to determine mutual contacts between social objects and, eventually, to create a social tie between those objects [48]. A SO is also responsible to search for a given data by analyzing all the information published by friend objects in the social object graph and to send queries to the SIoT platform or answering specific data requests. A SO can be requested to accomplish those functions by an external entity through a set of APIs called the *Social Hook*.

3.4 Application Deployment

As introduced above, the iSapiens platform allows the dynamic deployment of VOs and Agents during system runtime. Given a running set of iSapiens servers, an application can be deployed in all its software parts, by exploiting a *Deployer* entity, which is capable of sending agents and VOs code on the involved set of nodes and of instantiating and starting them. The Deployer is also responsible for the starting configuration of the application, made by establishing acquaintance relationships among agents and sending application-dependant configuration messages.

4 Application Domains and Examples

This section summarizes and overviews some carried out research activities and some significant applications realized by exploiting the concepts, the features and the capabilities that are at the basis of the iSapiens platform. All the reported applications fall in the domain of Smart Urban Ecosystems. They range from smart city applications [5] and smart offices [44] to smart drainage networks [49] and power cloud applications [50].

4.1 The Smart Street Cosenza Project

The project [5] was developed in the city of Cosenza (Italy) and it was devoted to the realization of a Smart Environment named *Smart Street Cosenza*. The target city area involved in the project is shown in Fig. 2a. Such area is located in the centre of Cosenza and cover: (i) a Bus Station, (ii) a square, (iii) the main commercial street of the city, (iv) part of one of the main driveways and (v) the area around a commercial centre. The goal is that of furnishing an IT infrastructure which provides services devoted to the real-time monitoring of the status of the involved areas. The infrastructure was designed so as to be further extended over time, either in terms of covered area or in term of provided features and offered functionalities.

The chosen area was instrumented by deploying in it a set of thirteen computational nodes (Fig. 2b) and seventy wireless sensors nodes (Fig. 2c). Each computational node is mains powered and consists of a Raspberry Pi mod.2 board hosting the iSapiens platform. Each sensor node is equipped with a battery embedded and is powered by a solar panel. Two types of sensors exist namely *type A* and *type B*. The former comprises noise, temperature, relative humidity and luminosity sensors. The latter comprise air quality sensors, measuring the concentration of CO, CO_2, NO and O_3. Sensors and computational nodes are connected by using a wireless network infrastructure.

Fig. 2 Smart Street Cosenza. **a** Target areas; **b** Computational Nodes; **c** Sensors nodes. Green locations host one *type A* node and one *type B* node, blue locations hosts only a *type A* node

Raw measures are collected by the computational nodes, then filtered and aggregated. Final data are sent to a remote server that stores them in a DBMS. The geo-referenced sensor measures as well as the aggregated information indexes such as Simmer Summer Index (SSI), Humidex and Air Quality Index (AQI) are made available to citizens through a web-based application. Such information can be used to determine if a given monitored area is comfortable and healthy. Moreover, the city administrators can monitor critical climatic or pollution events, and the research community, can use gathered data for further investigations and analysis. The realized system are up and running since December 2015, with only a few sensors failures.

4.2 Enhancing Smart Objects with Social Capabilities

The use and availability of Smart Objects within our cities is ever increasing. Such objects are often the basic building blocks used to realize Smart Environments devoted to improving our quality of life by offering advanced services and innovative applications. However, we are still far from permitting a fully exploitation of the potentialities and features of such objects. In fact, usually they have different roles and are usually deployed and managed by a multitude of different players. They fulfill specific goals in isolation and, very rarely, devices belonging to different administrative entities are able to cooperate and/or share information even if they coexist in the same physical space and/or produce very correlate informations. Given a specific smart environment, smart objects might be classified as follows [2, 4]:

- *Native objects*: are the objects purposely deployed for the creation of an application. They communicate by using a dedicated infrastructure, and it is assumed that they are managed by a single administrative entity, thus they are implicitly trusted.
- *Foreign objects*: are independently conceived and deployed with respect to the creation of a given application. Anyway, they share (either permanently or occasionally) their location with the native objects. Example of such objects are traffic monitoring cameras, Smart-meters, pollution sensors and the occasionally passing smart-vehicles.
- *External objects* are independently conceived and deployed with respect to the creation of a given application, but never share their location never get in touch with native objects. Some examples of such objects are: smart meteorological stations located outside the city, the components of the national railways monitoring system, the stations for earthquakes monitoring.

The social components of the *iSapiens* platform are exploitable to deal with the segregation among those categories of objects. More in particular, *Native objects* become able to opportunistically discover and access the *Foreign* and *External objects*. The social components of *iSapiens* are inspired to the Social Internet of Things (SIoT) which permits to build a social network connecting smart-objects. Social relations are created according the characteristics of the objects such as their model, location, movement pattern and purpose (see Sect. 2.3). The *iSapiens* platforms leverages that social network by enabling: (i) information sharing between *Native, Foreign* and *External objects*, (ii) service discovery for finding objects providing some given services, (iii) service composition and, (iv) trustworthiness management.

In order to give evidence about the effectiveness of the social components of *iSapiens* we consider a city block of 2 Km^2 and a reference application which is based on 4 smart objects placed at fixed positions in the area. The SWIM software [51] was used to simulate N, $N \in [100,\ 300]$, people moving within the considered area. Each person bring with him his own personal mobile device. All together, the devices represent the *foreign objects*. While people move, their carried personal devices enter in contact (we assumed a sensing range of 10 [m]) and begin to create their SIoT network. By post-processing the output of SWIM through custom software written in *MATLAB*®, we asses how the social network between the *Native*, the *External*, and *Foreign objects* evolved after [10, 12, 24, 36] hours from the simulation beginning. Specifically, according to [32] we consider that two objects establish a *Social Object Relationship* (*SOR*) [10] after having experienced one or more contacts for a cumulative contact time of at least 10 [min]. We further considered all the *Native objects* connected by a *Co-work objects relationship* (*C-WOR*) [10]. Figure 3 shows the cumulative distribution function of the distance, in terms of hops, between *native* and *non native* objects in the social network.

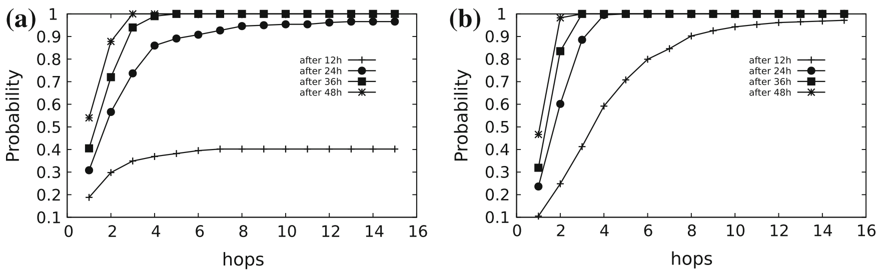

Fig. 3 Cumulative distribution function of the distance between *Native* and *Foreign* objects in the social network: **a** 100 nodes; **b** 300 nodes

From Fig. 3 it emerges that all the *Foreign* objects are included in the social network of the *Native* objects. As the time passes, the average distance in terms of hops between the *Natives* and the *Foreigners* get smaller, and, a shorter average distance in the social networks corresponds to a faster propagation and to a quick interaction between the objects. By comparing Fig. 3a, b it follows that the higher is the number of *Foreign* objects the faster is the inclusion process. Moreover, within 24*h*, 36*h*, 48*h* the inclusion process completed despite the number of the considered objects. As a final remark, we conclude observing that the *iSapiens* platform is really able in breaking down the segregation between smart objects belonging to different categories. More extensive results can be found in [4].

4.3 Urban Drainage Network

Urban floods are becoming more frequent as a result of the increase of impervious areas and of the occurrence of extreme weather conditions due to climate change. Moreover, obstructions and blockages, due to a poor maintenance of pipes and catch basins, may impair the hydraulic efficiency of an urban drainage system. Centralized stormwater measures [52], like the use of detention tanks and retention basins, can be exploited for managing urban flooding. However, those solutions are not easily applicable due to the lack of space, especially in densely populated areas.

The solution here considered [49], relies on a CPS which exploits the iSapiens architecture and its related concepts. The proposal consist in spreading in the drainage network a set of water level sensors along with some smart gates. Such devices are directly linked with a set of interconnected computing nodes. Computing nodes host a distributed and decentralized application based on cooperating agents whose aim is that of adjusting the gates in order to dynamically optimize the water load in all the conduits of the drainage network. The optimization of the water load makes the network able to fully exploit the storage capacity of the pipeline. In such a way, the

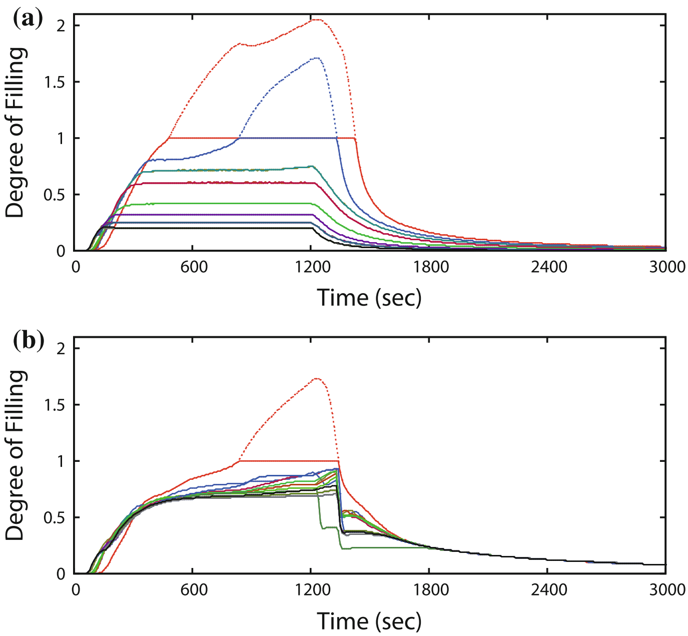

Fig. 4 Comparison of water loads (degree of filling) in different conduits, represented with different colors: base-line **a** versus controlled **b** scenarios. The areas above 1 and under the dashed lines represents the flooding volume of each conduit

exceeding stormwater volume is accumulated and hold within the pipeline itself thus avoiding (or limiting) a water overflow on the side-walks and street paving. Agents coordinate each other by running a gossip-based algorithm [53] devoted to compute the average of the water load. Such average value is then used by each gate-agent for tuning its gate so as to bring water loads closer to that average. This approach permits the systems to dynamically adapt also to unforeseen events occurring in the drainage network like damage, occlusions and so on.

The behavior of the system was assessed trough simulation. In particular, the EPA SWMM software [54], which is widely used in the hydraulic scientific community, was exploited in order to simulate the behavior of realistic urban drainage networks which include gates during severe rain events.

Simulation results proved that our algorithm, used as a real time controller for an urban drainage system equipped with a series of movable gates, is really able to manage flooding by exploiting the storage capacity of the less overwhelmed conduits of the system. Results are shown in Fig. 4. They demonstrate that the approach significantly reduces the flooding volume (the area between 1 and the dashed lines) in an urban drainage network during a heavy rainfall event. Other results can be found in [39].

4.4 An Open-Air Smart Museum

The *iSapiens* platform and related concepts were also exploited for the the realization of a smart museum in the city of Cosenza (Italy) [45]. The realized smart museum, namely the e-MAB smart museum, was developed by instrumenting and enhancing the MAB (Museo all'Aperto Bilotti) open-air museum.

The MAB is a permanent open-air museum which is located in the main commercial thoroughfare of Cosenza. The museum is located on the pedestrian area of Corso Mazzini, which is the main commercial street in the town comprehending shops, historical buildings, and cafes. The museum collects some prestigious sculptures of artists like Salvador Dalí, Giacomo Manzú, Sasha Sosno, Giorgio De Chirico, and by some other Calabrian artists. It was born thanks to the donation of Carlo Bilotti, a wealthy collector native of Cosenza but immigrant to America. Figure 5 reports the location of the MAB along with the displacement of the artworks in it.

The primary objective of the e-MAB is that of strengthening both safety and security of the artworks. The basic idea behind the smart museum is equipping the statues with "virtual senses" and turning them into *cognitive entities* which are *aware* of what is happening in their neighborhood so as to discover/recognize/react to potentially dangerous situations. Each statue has been purposely enhanced with a set of sensor/elaboration/actuator devices enabling the perception of the surrounding environment and the execution of deterrent actions (e.g., asking for help). Physical devices have been deployed in safe places close to the statues in order to avoid artwork damages. More in particular, the *virtual sight* has been implemented through cameras, the *virtual hearing* through microphones, the *virtual touch* through IR proximity sensors, and the *virtual speech* through speakers. Virtual senses were used to infer when a situation become dangerous. For instance, a situation is considered as dangerous when a statue is hit or when too many visitors are close to the artwork. Deterrent actions comprehends light flashing at the statue and the start of recorded alert messages on the speakers.

The second objective of the e-MAB is to augment its attractiveness and enjoyability with respect to visitors. Some services were developed for (i) enabling visitors to automatically take a "selfy" with an artwork, (ii) obtaining detailed info of an artwork, (iii) gathering and making it available information about the "comfort" level of

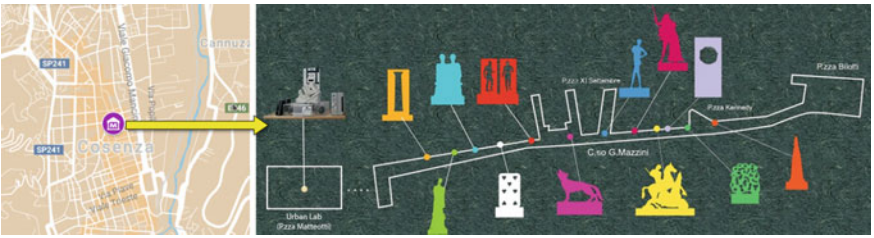

Fig. 5 The MAB location and a map of its artworks

the museum (in terms of crowding and climatic wellness), (iv) promoting information exchange among the town administrators and visitors (alert about strikes, public service interruption and so forth), commercial activities and visitors (for advertisements and promotions), and visitors and visitors (virtual post-it service). For this second purpose, a web-based application has been developed.

4.5 A Smart Office Implementation

A smart office prototype [44], with the aim of monitoring environmental conditions and applying some simple actuation rules in order to avoid waste in energy consumption, was implemented at ICAR-CNR.[2] The instrumented office room contains (i) three working desks having some computers and lamps on them (ii) one chair for each desk and another set of chairs for meetings (iii) a big window on the east side of the room. The fundamental features of the *isapiens* platform were exploited for the realization of the prototype.

From a design point of view, a first step in developing the system was devoted to defining the functionalities of the smart office. Subsequently, such functionalities were related to the physical devices needed to support them. From Fig. 6 it follows that three main functionalities were considered: (i) Monitoring RoomCondition, (ii) Monitoring DeskCondition, and (iii) Controlling DeskCondition. The latter is in charge of recognizing energy waste at the desk (e.g., if no one is at the desk and the lamp is on) and turn off the lamp if required. In the figure, the basic functionalities used to offer the main ones are listed. Moreover, for each basic functionality, the required physical devices are shown. The DeviceAbstractor is an associative class, which mirrors the use of virtual objects required for managing heterogeneity issues.

For implementation purposes, a set of three physical sensor nodes consisting in Waspmote nodes[3] (see Fig. 7) were deployed in the office: one for the desk, one for the lamp, and one for the room environment. The sensor node at desk hosts a Presence Infrared (PIR) sensor coupled with a sound sensor (for monitoring the activity at the desk), and a luminosity sensor. The lamp is instrumented with a smart plug sensor node, having an electricity sensor and a relay actuator, for monitoring and controlling the status of the lamp. The room sensor node hosts a PIR, a sound, a luminosity, a temperature, and a humidity sensor. Each sensor node sends every 30 s the average of sensed data with respect to a 30 s time window.

Two *iSapiens* servers were deployed in the office on two dedicated Raspberry Pi[4] single-board computer. One server was for the desk (linked to the desk and the lamp sensor nodes) and one was for the room (linked to the room sensor nodes). Communication among sensor nodes and Virtual Objects relies on the zigbee protocol stack.

---

[2] ICAR-CNR, via P.Bucci, cubo 7-11C, 87036 Rende, Italy.

[3] http://www.libelium.com/products/waspmote/.

[4] https://www.raspberrypi.org/.

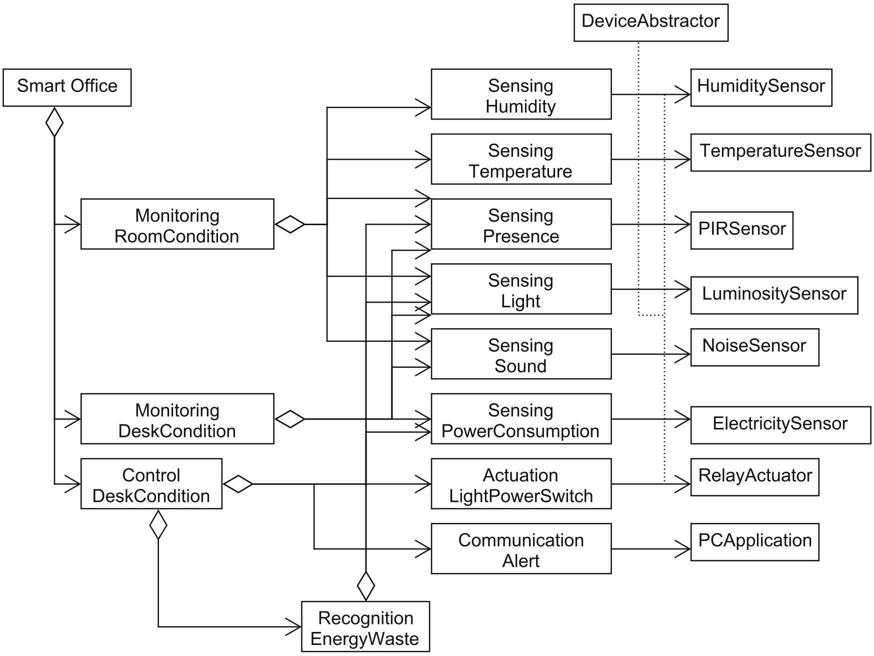

Fig. 6 A UML class diagram modelling the smart office

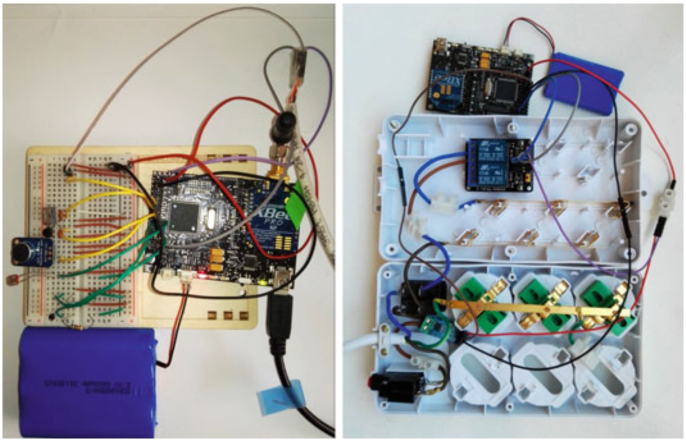

Fig. 7 The room sensor node (left) and the smart plug sensor node (right)

Accordingly to the *iSapiens* features, for each functionality a dedicated agent was developed for implementing it. The agents are deployed on the *iSapiens* server according to the closeness to the physical devices an agent has to manage. Some further Off-network services were also implemented to realize the persistence of gathered data and to signal alerts on energy waste to user clients.

The realized system is extensible as the agent servers can be further added and populated so as to furnish new functionalities which can also be obtained by composing the exiting ones. Scalability is supported, as new nodes can be deployed in the same or other offices. Fault tolerance is fostered because the computation is carried out locally in each node, and the crash of a single node does not compromise the other ones. The system is running since the middle of 2016.

4.6 Implementing e-Health Personal Services

The population of the most developed countries is constantly aging. As a consequence, it becomes of utmost importance the development of e-Health systems devoted to improve the Quality of Life of suffering/elder people. In this context, the Internet of Things (IoT) paradigm [55], proved to be effective in determining a sort of evolutionary leap in e-Health systems [56].

Despite the huge potentialities arising from the exploitation of IoT-based technologies in the domain of e-Health systems, the full integration between IoT and E-Health systems is still far from an actual implementation. Iot-based systems are usually made by a multitude of devices which are extremely heterogeneous and are deployed by single and specific users and companies. As a consequence, it becomes very difficult to estimate how many objects are available in a given area, what are their capabilities, and how they can jointly be used to solve a given problem. Moreover, as an additional constraint, the managing of integrated IoT and e-Health systems should be as transparent as possible with respect to the patients that should use them. In fact, a patient is usually a suffering person who is not to be troubled with any further complications related to the use of e-Health systems. The failing of this latter requirement could paradoxically cause a reduction in the Quality of Life of the patients.

The social components of the *iSapiens* platform give us the abilities to develop a *Personal Health Gateway* (*PHG*) (see Fig. 8) able to become the link between the IoT and e-Health domains. According to similar approaches in the literature [57], the *PHG* is a portable device usually carried by the patient having the task of collecting Health related measures from the patient and to make that information available for the Health related applications, which, in our case are implemented as *iSapiens* agents. A *PHD* is equipped with two interfaces: (i) an *E-Health* interface devoted to collect measures coming from the biomedical sensors worn by a patient, (ii) a *IoT* interface toward the *iSapiens* platform. While the latter interface is specific of the *iSapiens* platform, the E-Health interface is based on the specification of the Continua Health alliance [58] and hence there are no compatibility issues with the

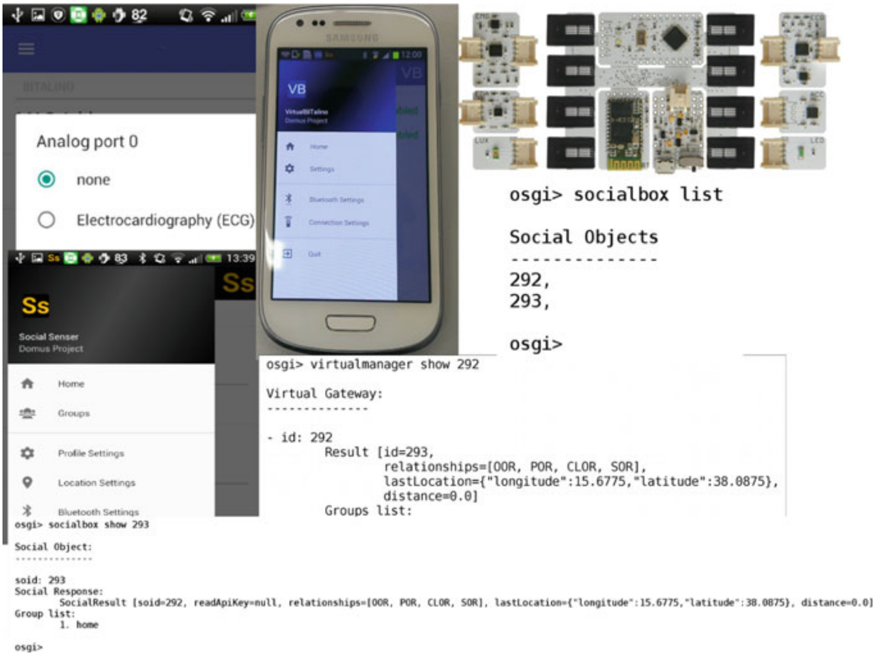

Fig. 8 The Personal Health Gateway (PHG), some snapshots of the user interface and a sample log of the correspondent Virtual Object on the *iSapiens* Platform

legacy E-Health devices. The *PHG*, as an iSapiens object, is able to discover other similar objects surrounding the patient and to create with them social ties according the rules of SIoT [10]. Hence, the medical applications running on the iSapiens platform might therefore access and process all the information made available by objects in the social network of the *PHG*.

As a case study, we implemented a prototype *PHG* on the basis of a commercial Android smartphone (see Fig. 8). We connected the smartphone to a BITalino board[5] through its Bluetooth interface in order to collect and acquire biometric data from a patient. Moreover, the *PHG* has been retrofitted with two customized software modules in order for it to become connected to the *iSapiens* platform. The first module encloses the acquired data from the Bitalino on-board sensors into properly formatted JSON messages, and forwards them to the *iSapiens* platform. The second module, the Social Senser, periodically overhears the transmission over the Wi-Fi and Bluetooth interfaces to search for other nearby social-enabled devices. This information, once forwarded to the *iSapiens* platform, is used to determine mutual contacts between social objects and, eventually, to create a social tie between those objects. Simulations confirmed the effectiveness of the social approach in dynamically discovering new

[5]http://www.bitalino.com/.