SEMANTIC TECHNIQUES FOR IOT DATA AND SERVICE MANAGEMENT: ONTOSMART SYSTEM

L. Nachabe [3], M. Girod-Genet [1] and B. ElHassan [2]

[1]Department of Réseaux et Services Multimédia Mobiles, Telecom SudParis University,
Evry, France
[2] Department of Electricity and Electronics, Faculty of Engineering, Branch 1,
Lebanese University, Tripoli, Lebanon.
[3] American University of Culture & Education, Faculty of Science,
Beirut, Lebanon.

ABSTRACT

In 2020 more than50 billions devices will be connected over the Internet. Every device will be connected to anything, anyone, anytime and anywhere in the world of Internet of Thing or IoT. This network will generate tremendous unstructured or semi structured data that should be shared between different devices/machines for advanced and automated service delivery in the benefits of the user's daily life. Thus, mechanisms for data interoperability and automatic service discovery and delivery should be offered. Although many approaches have been suggested in the state of art, none of these researches provide a fully interoperable, light, flexible and modular Sensing/Actuating as service architecture. Therefore, this paper introduces a new Semantic Multi Agent architecture named OntoSmart for IoT data and service management through service oriented paradigm. It proposes sensors/actuators and scenarios independent flexible context aware and distributed architecture for IoT systems, in particular smart home systems.

KEYWORDS

Smart-Home, IoT, Multi-Agent, distributed systems, WSN, semantic, Ontology, sensing as a service, interoperability, semantic interoperability.

1. INTRODUCTION

The continuing rise of mobile internet access, especially the emergence of 5G technology, in addition to the significant increase of smartphone users and the integration of mass scale cloud computing, are now transforming our society to what is called Smart Society. C. Levy et al. defined Smart Society as "the potential of digital technology and connected devices and the use of digital networks to improve people's lives[1]. Smart Society, in particular smart cities and smart homes, are formed of wireless sensor networks capable of sensing, communicating, computing and potentially actuating. These wireless networks can play the role of decision makers to facilitate the life of users. For example, when temperature exceeds a threshold the

DOI: 10.5121/ijwmn.2016.8403

window is opened automatically. The solutions are really endless. In fact, the Smart Society affects all aspects of our life including health services, smart home and automated/remote controlling and monitoring services, smart utility consumption (energy, water) and smart grid, smart mobility and Vehicle Area Networks, smart public services (parking, safety), Intelligent Transportation Systems, smart entertainment, etc.[2]. All of these solutions are providing remote monitoring and control capabilities (in particular through 5G technologies) using machine to machine communication (M2M) for intelligent data interpretation, new data and knowledge creation and fast intervention. Fortunately, it will pave the way to the development of new business and services opportunities. Industry analysts predict that 50 billion devices will be connected to mobile networks by 2020 [3]. University of Edinburgh is involved in a project call Smart Society which objective is “to move towards a hybrid system where people and machines tightly work together to build a smarter society”. They stressed on the importance of bridging the semantic gap between low-level machine and high-level human interpretation of data in order to collaborate for conflict resolution goals both at individual and societal levels [4].

5G Era is empowering the idea of stay connected, which means that everything is connected to anything, anyone, anytime and anywhere. This what has been dubbed the Internet of Thing (IoT) [5]. IoT is formed of extremely heterogeneous nodes in terms of roles (sensor, actuator, relay, etc.), manufacturer, communication interface, data rate, data type, etc. Moreover, these nodes can have limited resource capabilities (memory, battery lifetime and CPU), where tremendous data is generated. That is why, in most cases, these generated data should be collected and treated in an external node as local/web server or cloud server [2]. Cloud solutions have been integrated in these systems because of its unlimited capacity of storage, processing power, virtual resources creation and service delivery.

X. Sheng et al. [6] and A. Zaslavsky et al. [5] introduced the idea of sensing as service. With the growth of cloud computing the idea of offering everything as service has been integrated in almost every business model. Sensing as a service (SeaaS) reveals the idea of transforming physical word (sensors/actuators) to a service that can be discovered and invoked by users on demand. To develop SaaS solutions, developers should procure on-demand services, data portability and interoperability between different data providers, context and situation awareness as well as mechanisms dedicated for security and privacy.

In this paper, we present a new general architectural for data collection, processing and management of IoT systems. Furthermore, this architecture paves the way towards SaaS paradigm where users can request any discovered service regardless the underneath mechanisms (data collection and sensor configurations). The architecture relies on semantic techniques to tackle the problem of heterogeneity within the same system, and between different data systems. Furthermore, it uses the idea of web semantic web servers to provide SaaS solutions. In addition, it is based on multi-agent architecture in order to distribute the processing among different components. This leads to flexible, embedded devices compatible and more power consumption efficient solution. All these aforementioned ideas will be presented in the next sections.

The paper is organized as follow. In Section II IoT challenges are described in order to explain the reasons behind using ontologies and multi-Agent architectures. Overview of the retained modular ontologies is also presented in this section. Section III describes the general architecture of our proposed system called OntoSmart system describing the role of different agents. Section IV details our OntoSmart based applications called OntoSmartHome for data monitoring and management. Moreover, tests and analysis are given in this section. Finally, Section V concludes the paper.

2. BACKGROUND

2.1. IoT Challenges

M. Truck defined the IoT “the transformation of any physical object into a digital data product” [7]. In other words, using your smartphone as pedometer in order to calculate your burned calories, and publishing these data on social media is part of IoT. In fact, we are now part of IoT which is becoming a real booming topic. The miniaturization of sensors, the constant evolution of wearable devices, the connectivity anywhere at any time and the use of smartphone are conducting us to be part of the IoT. Although IoT paradigm is fundamentally controlling huge amount of our daily activities (calendar notification, social media, health applications, cloud storage applications, weather broadcasting, search engines, etc.), but still have paramount challenges.

The European Research Cluster on IoT (IERC) defined five challenges [8]:

•Develop reference architecture to enable cross industry technology cooperation and resolve the problem of interoperability.
•Provide complex services and composed solutions.
•Integrate heterogeneous technology at various levels.
•Combine various business models.
•Procure secure, private and reliable solutions.
Zaslavsky et al. insisted on the importance of providing a flexible, easy and distributed architecture for IoT where data is provided as service. Moreover reasoning and decision making should be integrated in the architecture [5].

Texas Instruments [9] presented the main challenges of IoT as follow:

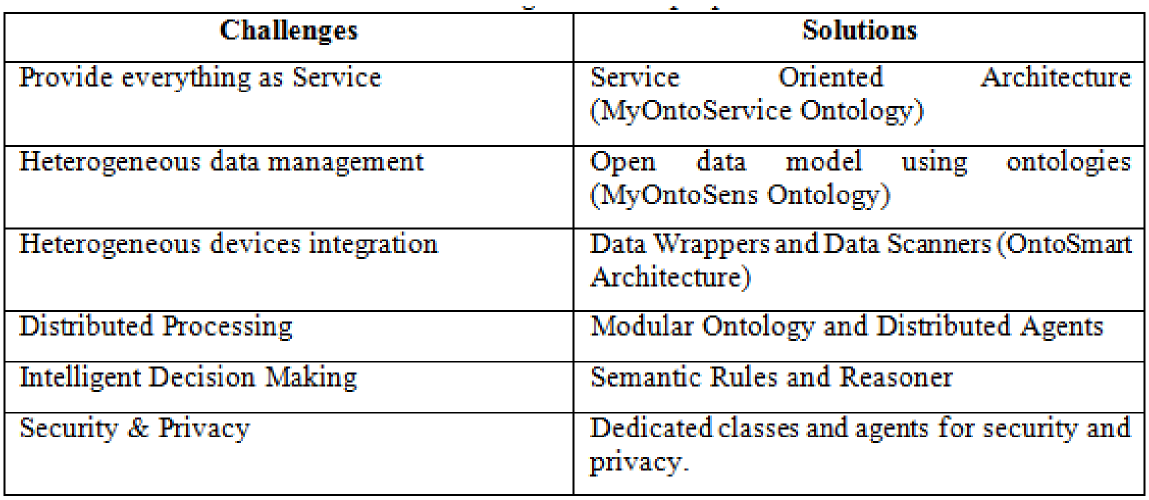
•Sensing a complex (Heterogeneous) environment.
•Wide variety of wireless/wired communication protocols.
•Critical power consumption.
•Publishing data to the cloud.
•Security concerns.

Texas instruments summarized the IoT into three levels: the physical devices, the IoT gateway/relay and the cloud level.

Where billions of devices have been connected, IoT is characterized by Big Data[5]. Volume, velocity and variety are at the core of big data. Thus, the main challenges of IoT are how to process this high volume of processing with low power devices. Moreover, the data should be analyzed in real time (especially for health services where immediate intervention should be done) and enriched with social media options (people nowadays are familiar with social media). Although data can be shared, security and privacy concerns are still considered the main question in IoT.

Table 1 summarizes the points that should be respected when designing IoT solution. Moreover, Table 1 depicts how our proposed IoT architecture addresses each of these points. In the next sections, all these points are detailed.

Table 1- IoT Challenges and our proposed solutions



Challenges	Solutions
Provide everything as Service	Service Oriented Architecture (MyOntoService Ontology)
Heterogeneous data management	Open data model using ontologies (MyOntoSens Ontology)
Heterogeneous devices integration	Data Wrappers and Data Scanners (OntoSmart Architecture)
Distributed Processing	Modular Ontology and Distributed Agents
Intelligent Decision Making	Semantic Rules and Reasoner
Security & Privacy	Dedicated classes and agents for security and privacy.

2.2 .Reason Behind Using Ontologies

As IoT encompasses different systems, applications, and WSNs, different terminologies are used to describe the same properties or object. Moreover, various ranges of sensors/actuators (ambient sensors, electrical devices, biomedical sensors, etc.) from wide range of vendors supporting different wireless technologies (Bluetooth LE, LoRa, ZigBee, etc.) are used [7]. This variety caused the problem of heterogeneity management and interoperability. In general, these systems are monolithically deployed. The integration of a new device or a new service needs the reconfiguration of the overall system, and some time the deployment of a new system [8]. For that reason, new techniques that deal with the data meaning and enhance the automatic node and service discovery are needed. These techniques are named Semantic techniques where an entity presents an aspect of the real domain described by metadata (data about data). Thus, by using semantic techniques within IoT, the raw data can be annotated by semantic data respecting a predefined semantic model to unify its description [9]. In that way, the problem of interoperability between different systems can be resolved. Ontologies are the most powerful semantic techniques. Using the formal definition, ontologies are "tools for specifying the semantics of terminology system in a well-defined and unambiguous manner" [10]. W3C recommended the use of OWL language to describe ontologies [11]. In fact, the most powerful point behind using OWL is the capability of reasoning in order to infer more significant data.

Many researches have introduced the usage of ontologies in IoT. The IoT-A project presents a high level abstraction of the IoT domain. It basically consists of a Domain Model, Information model and Communication Model [10]. The Domain Model includes a user who can be a Human or an Active Digital Entity. The user interacts with a physical entity and invokes at least one service. A virtual Entity can be associated with a service or/and a resource. The service accesses at least one resource. The resource can be storage, network, or on-device resources (which hosts a device). A device can be an actuator, a sensor or a tag device. The authors in [11] extended the works done in the IoT-A project by adding semantic description and linking the models to existing domain specific ontologies. The core of the IoT, Information model, consists of Entity, Resource, Service and Device.

Almost all IoT researches suggested including the idea of sensing as service. Thus, these researches use OWL-S service ontology as upper service layer. OWL-S [12], the semantic mark-up for web services, developed by DARPA DAML program using OWL languages aims to describe semantic web services. It permits the automatic web service: discovery, invocation, composition, interoperation, and execution monitoring. The OWL-S ontology includes mainly three sub-ontologies: the service profile ontology that describes what the service does; the process model ontology that describes how the service is used; and the grounding ontology that describes how to interact with the service. Nambi et al. [13] proposed a unified semantic knowledge for IoT where resources are readable, recognizable, locatable, addressable and/or controllable via the Internet. The Resource Ontology describes the sensors, actuators, physical objects and composite objects. It is reused from the SSN ontology [14]. The Location Ontology adds the geospatial information to IoT information. It is reused from the GeoNames ontology [15]. The Domain ontology models a specific or a generic domain such as E-health, smart home, etc. Any domain ontology can be imported depending on the IoT application. The Context Domain enables context-awareness and contextual interoperability during service discovery and composition. Aspect-Scale-Context (ASC) is used to model conceptual information. The Policy Ontology defines how to accomplish a service requested by a user. There are three different policy levels: high level policy defining abstract service, concrete policies for specific services, and low level policies defining the execution plan of the services. The Service Ontology is built upon the concept of OWL-S.

Wang et al. [16] introduced a new Semantic Model for IoT architecture composed of seven main modules. The IoT resources are defined by extending the SSN ontology to include actuators, servers and gateways. The Observation & Measurement Ontology is reused from the SSN Ontology. The entity of Interest and physical locations represent the object of the physical world.
It is rigorous that none of these attempts combine the fully description of the sensors/actuators, data, process, and services. While reliability and service consistency are primordial in such systems, the QoS and computational resources as well as remaining battery level are needed for service modeling. In addition, IoT involves wide range of data/service accessing protocols (HTTP, CoAP, MQTT, SOAP, etc.). Thus grounding methods for service accessing will facilitate the service discovery.

Therefore, the contribution of this work is OntoSmart system that relies on both MyOntoSens Ontology [17] (summarized in Figure1) and MyOntoService Ontology (depicted in Figure 3) to model IoT data and services. These ontologies are provided with associated management enablers, within our OntoSmart system, and will enable the handling of all the aforementioned IoT challenges cited in Table 1.

3. ONTOSMART SYSTEM ONTOLOGIES

IoT can be seen as a WSN (Wireless Sensor Network) encompassing different nodes (sensors, actuators, equipment, servers and gateways). Each WSN (modeling a home, building, or network) can have one or more owner (e.g. the patient to be monitored at home) and contact persons (medical staff or relatives). Each node is used for certain process (temperature, humidity, etc.) and each actuator performs certain action (turn ON/OFF, shuttle Up/Down, etc.). That is why we used MyOntoSens ontology detailed in [17] and extended in [18] to describe the components of our proposed system since this ontology already takes into account all the required attributes that model WSN (network description, process, data, constraints, measurements, sensors, actuators, etc.) . MyOntoSens has been designed as a modular ontology and is divided into three parts: MyOntoSensWSN, MyOntoSensNode and MyOntoSensProcess. Figure 1 summarizes the main classes of MyOntoSens ontology, while Figure 2 depicts the main classes and attributes of MyOntoSensProcess part.

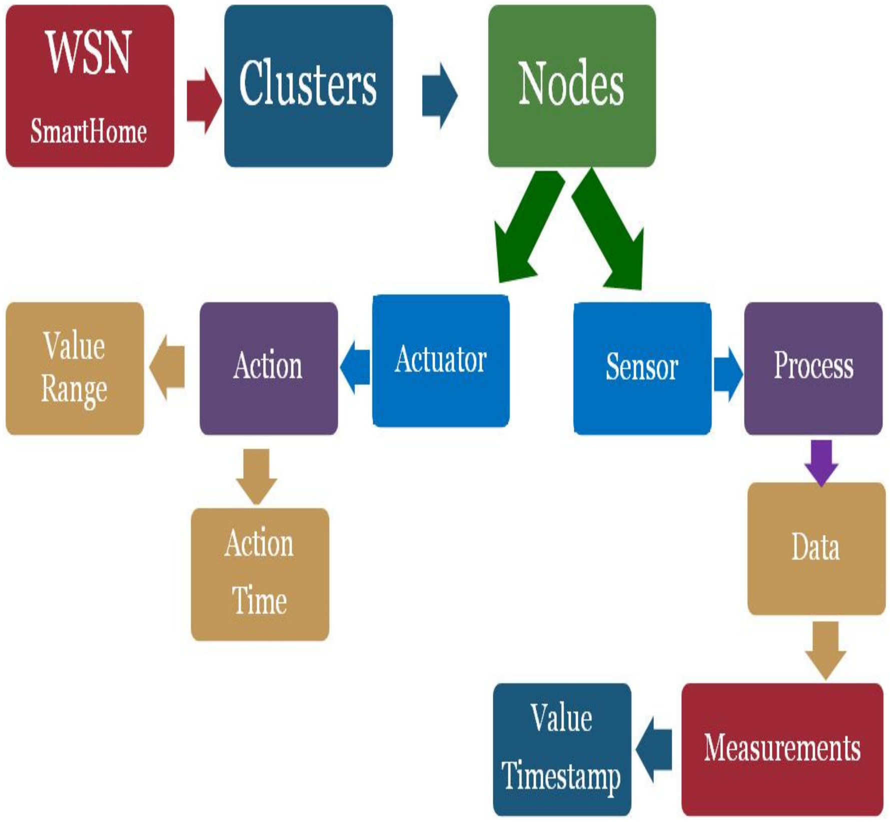


Figure 1-MyOntoSens Ontology

Moreover, the IoT should provide services to the users (monitoring, notification, reminder, etc.).

We are modeling these services by introducing original service ontology, called MyOntoService

•MyOntoServiceProfile that describes the service (QoS, QoI, service constraints and ontology and depicted in Figure 3. Our new upper service ontology adopts the idea of OWL-S ontology [18]. It is composed of three main sub-ontologies (as depicted in Figure 3):service type) and enables their auto-discovery,

•MyOntoServiceProcess that describes how the service is realized based on the Input, Output Precondition, and Effect (IOPE) mechanism,

•And finally MyOntoServiceGrounding that describes how the service can be invoked (or accessed in IoT e.g. HTTP, CoAP, etc.)

To ensure automatic senor/service discovery, the ontologies are described by SWRL [18] rules and the Pellet [19] reasoner is used to infer explicit data. MyOntoService ontology will enable the use of Sensing as Service paradigm due to the use of the semantic rule (Rule 1). When a sensor is used for a process, this process will be automatically inferred as atomic service enabling the automatic sensor discovery. The user can enable/disable the service by adding the object property “enable”.

Rule 1: Process(?P) -> Service(?P); Sensor(?S), Atomic(?a), usedFor(?S, ?a) -> hasProducer(?a, ?S)

4. ONTOSMART MULTI-AGENT ARCHITECTURE

The main aim of our proposed system is modularity, scalability, flexibility and distributed computing. Thus, this architecture should logically be divided in standalone modules offering certain services, capable of interacting with other modules and that can be instantiated when required. This is exactly what a software agent can do. An agent is a piece of software that can run autonomy to perform certain behavior. The composition of many agents is called Multiple Agents System (MAS) [20]. The agents interact by passing predefined messages. The MAS architecture describes how different agents are interconnecting. MAS are widely used in complex system where abstract definition of certain task is required. Figure 4 depicts this multilayered architecture. In our proposed system basic functionalities are offered by agents capable of invoking other agents. This is in particular the case for our data collection and information management/sharing/retrieval functionalities. For example, we can cite the data collector agent, query agent, notification agent, writer agent, etc.

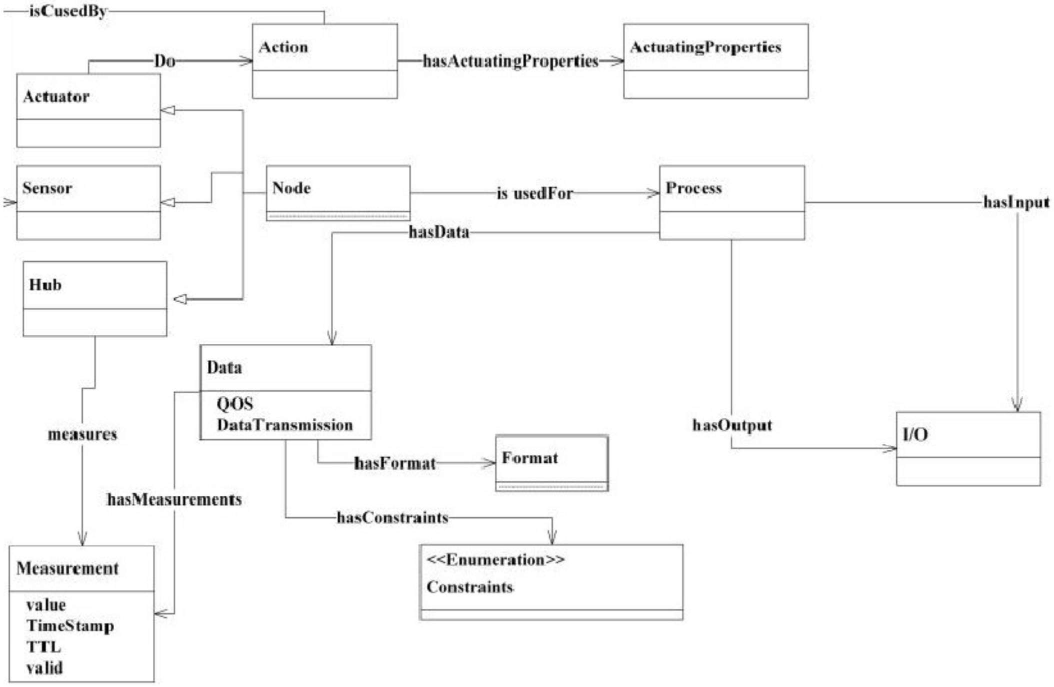


Figure 1-MyOntoSensProcess Classes Hierarchy

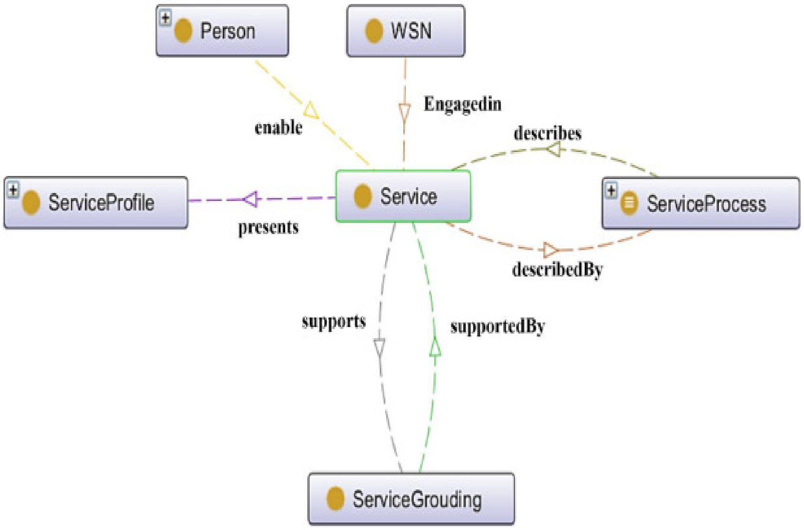


Figure 2-MyOntoService Ontology

Moreover, using expressive ontologies without defining a general mechanism for data annotation is pointless [5]. That is why we defined three main agents in our overall system:

•Scanners: for each communication protocol (Bluetooth LE, WIFI, ZigBee) there is a data scanner that read raw data and call the adequate wrapper for semantic annotation. Moreover, for each grounding protocol (CoAP, HTTP, etc.) there is a service scanner that reads the request from external user and calls the adequate service.

•Writers: To send data from the system to external users, the adequate writer encapsulates the sent data based on the requested protocol.

•Wrappers: To add semantic description of the raw data based on MyOntoSens and MyOntoService ontologies.

Detailed description of how to use these components is depicted in section 4.

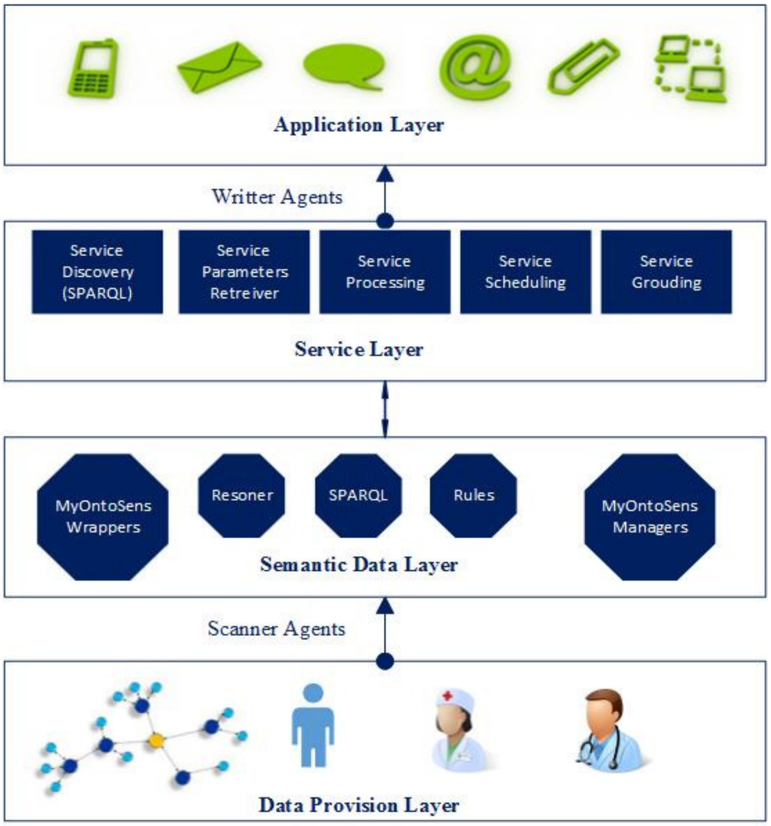


Figure 3- OntoSmart General Architecture

4. IMPLEMENTATION OF ONTO SMART SYSTEM ONTOLOGY

The increasing number of older persons living independently, in parallel with the considerable number of individuals with chronic diseases or disabilities pushed the researchers to work on the idea of smart home systems [18]. These systems help to keep individuals safe, to assist them in controlling home appliances, and to inform their relatives and the medical staff about their status. Thus, to test our OntoSmart architecture we have chosen the case of a smart home dedicated for elderly monitoring and assisting. This use case is offering sensors/actuators and scenarios independent flexible context aware and distributed solution based on standardized WSN and service ontologies as well as multi-agent architecture described in sections 2 and 3. Figure 5 depicts this smart home.

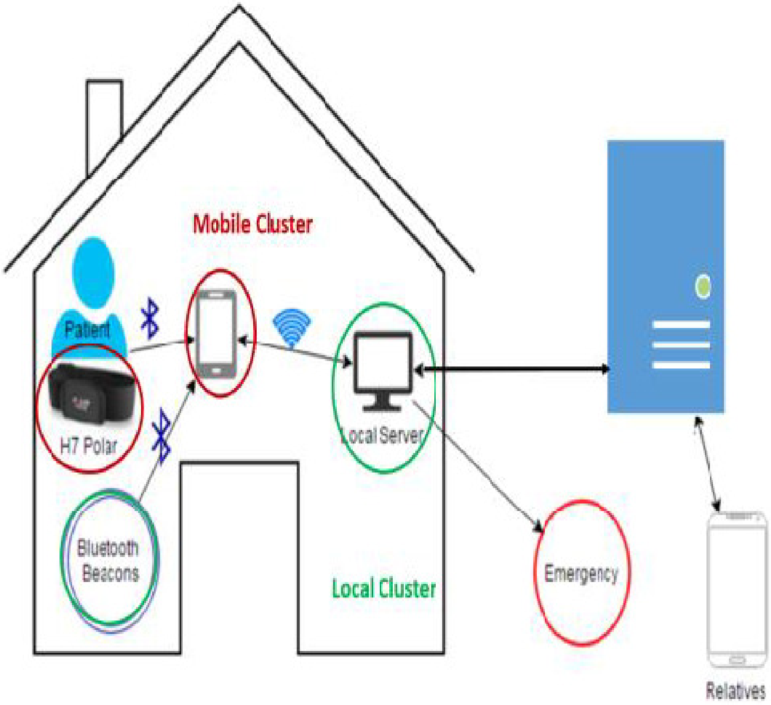


Figure 4- OnoSmart Home Clusters

4.1 WBAN Cluster

The main aim of the WBAN Cluster is to collect patient's vital signals in order to send it to the local server. Here comes the role of the Data Collector Agent (DCA), also called the Cluster Hub. In our scenario, the WBAN Cluster is formed of the heart rate sensor, acceleration sensor and the smart phone of the user which is playing the role of the Cluster Hub. To capture the heart rate, the H7Polar sensor is used. This senor sends the heart rate via its Bluetooth LE interface.

Therefore, a Bluetooth LE Scanner (BLES) agent is installed on the patient's smart phone in order to:

•identify any Bluetooth LE device,

•and recognize the offered service due to the use of UUID defined in standard GATT Profiles [21].

This scanner gives the first step toward an automatic sensor discovery.

To determine the posture of the patient, the smart phone built-in three acceleration sensor is used. The detail of the posture calculation is given in [18].

After identifying the node's characteristics, these latter should be sent to the local server for semantic annotation and further treatments. JSON [22] message format was adopted due to its lightness and simplicity. Therefore, a JSON Writer (JS) agent is installed on the smart phone in order to encapsulate data and send it to the local server.

If other types of sensors are used (like ZigBee), the only think that should be done is to install the convenient scanner (like ZigBee Scanner). Likewise, if data from the smartphone is encapsulated using a message format other than the JSON messages, the suitable writer agent should be compelled. In fact, the use of these data agents and writers enable the automatic discovery of any added nodes and enhance the flexibility and scalability of the system where any nodes can be added without the need to rebuild the overall system. What is really required is to ensure that the communication protocol used by the device has the suitable data scanner agent.

4.2 .WSN Cluster

The WSN Cluster is formed of the sensors dedicated for indoor localization, the ambient parameters sensors, and the local server (DCA or cluster Hub). In our case, we use the RadBeacon Bluetooth LE sensors to identify the location of the patient. Moreover, the ambient parameters (light intensity, humidity and temperature) are retrieved from the built-in sensors in the patient's smartphone in order to be sent to the remote server using JSON messages. We installed, on the local server, the following agents:

•BLES agent: to identify Bluetooth LE sensors and retrieve the required information in order to invoke the adequate wrapper.

•JSON Scanner (JS) agent: to retrieve data sent from the smartphone and invoke the adequate wrapper.

•Node Wrapper (NW) agent: to add semantic annotation about the sensors/hubs based on MyOntoSensNode ontology.

•Process Wrapper (PW) agent: To add semantic annotation about the process used by the sensors based on MyOntoSensProcess.

In that way, the local server will contain all the semantic description of the home devices. Because more than one patient can use the same WSN sensors (e.g. indoor localization sensors can be used to locate more than one patient in the same home), the object property "measuredFor" has been added between the sensors measurements (Measurement class of

MyOntoSensProcess ontology) and the patient (Patient Class of MyOntoSensWSN ontology). To this point, all nodes are automatically discovered. Regarding the service discover, we added SWRL [12] rules to infer basic services offered by the system. All processes are considered as service (e.g. heart rate, temperature, etc.). This inference is insured by using Rule 1. Composed services are deduced from additional rules. For example, using the heart rate sensor implies that the maximum heart rate service can be used for notification purposes (Rule 2).

(Rule 2) Process (HR) -> Service (MaxHR), hasEffect (MaxHR, Notification).

Moreover, the object property "enable" between patient individual and service individual is used for allowing the patient to enable/disable any service offered by the system. Only enabled services are retrieved when invoking the query dedicated for service discovery.

4.3 Remote Semantic Storage and Management Server

The main aim of the remote semantic server is to save the fully semantic description of the smart home. It plays the role of semantic registry. While intelligence treatment for each home is located on its local server, the intelligent cooperation between different TSHCSs, and advanced data analysis for statistical purposes can be conducted on this remote server. In our scenario, the remote server encompasses the following agents:

•SPARQL Agent: to query semantic information saved on the remote/local server in order to retrieve new semantic information.

•CoAP Agent: to publish the data in order to be used by external users for remote monitoring (e.g. patient's relatives and medical staff). The CoAP [23] is a Rest-full protocol used for constraint networks. It uses the UDP protocol as transport layer and the reliability mechanism is driven by the application layer. We used the CoAP Confirmable messages to ensure the required reliability in our proposed TSCHS system. In that way, the Remote server will play the Role of CoAP monitoring server allowing remote users to access the data by using CoAP GET requests.

•Notification Agent: to notify external users about abnormal patient's attitude and dangerous measured values. In our case, we notify the user if the patient is falling or if his/her heart rate exceeds the normal value. It is worthy to note that the security mechanism is not within the scope of this study. That's why we based our test-bed authentication mechanisms on Android GCM due to its implementation simplicity, and by assuming that all our system users have a Gmail account. Let's precise here that for a commercial application, stronger authentication mechanisms will obviously have to be designed and implemented. Figure 6 depicts all the aforementioned agents and building blocks location and interactions.

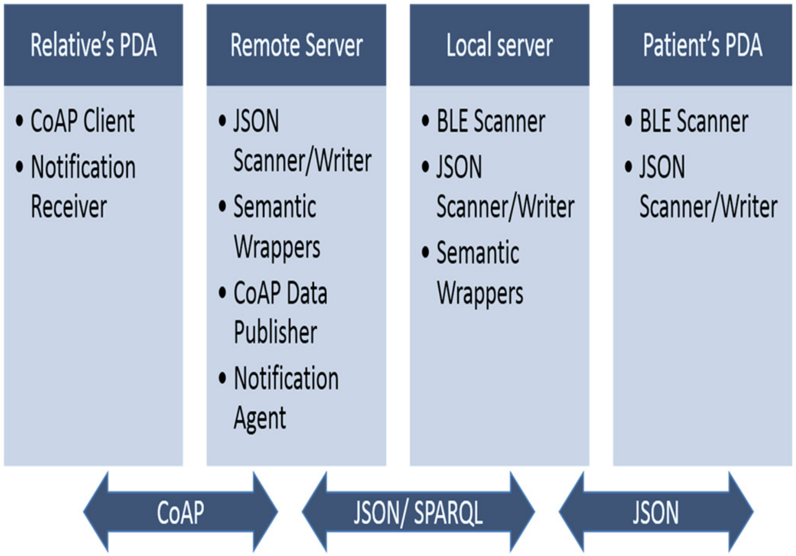


Figure 5-Implemented Agents and interactions as deployed in OntoSmart System

The remote server will invoke the WSN Wrapper in order to add the new WSN and Patient into its semantic registry. The PatientID (identifier generated by the remote server) is now returned to the smartphone and saved in its shared Preference (local database on the Smartphone) for being used when needed (for patient identification). The patient can now launch its OntoSmartHome application and log into the OntoSmart system using his/her Gmail account.

5. ONTOSMARTHOME APPLICATIONS

Two applications have been developed for OntoSmart system: the first application (detailed in Section A) is dedicated for the patient in order to initialize the system and display the monitoring values; the second application (detailed in Section 5.2) is dedicated for relatives for remote patient's monitoring. The same monitoring interface (depicted in Figure 7) is retained for the patient and relative applications.

5.1. Patient's OntoSmartHome Application

We have provided our OntoSmart system with a patient-dedicated application and corresponding applicative agents (see Figure 6). This application, depicted in Figure 7 and Figure 8, is integrated within the Smartphone of the patient. The details of the patient's mobile application can be found in [18]. Note that a compromise between system accuracy and battery consumption is mandatory in such systems. That's why, after practical testing, we have chosen to acquire 50 samples/sec from the sensors. In addition, these modified values will be sent to the local server and the remote server via the JW in order to be updated on the CoAP server and registered in the remote ontology repository.

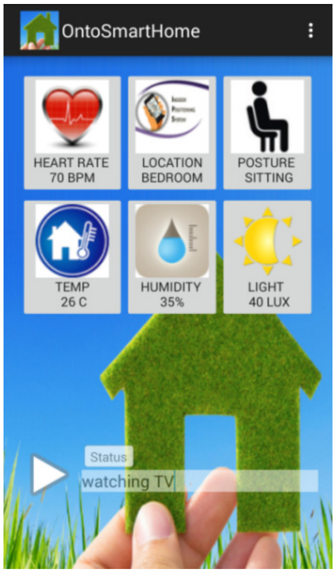
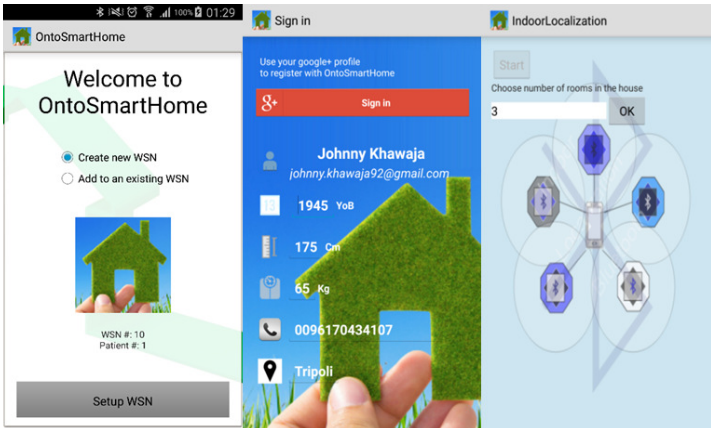


Figure 6- Monitoring Interface of OntoSmartHome Application



Johnny Khawaja
johnny.khawaja92@gmail.com
1945 YoB
175 Cm
65 Kg
0096170434107
Tripoli
3

Figure 7-Patient's OntoSmartHome application

5.2. Relatives OntoSmartHome Application

The first time the application is installed, the relative should successfully log in using his/her Gmail account. Afterward, the relative is asked to enter the patient's email. The patient and relative emails are sent via the JSON Writer (JW) to the remote server where a SPARQL query is invoked to verify the relative permissions. Figure 8 depicts the overall authentication process for the relative's mobile application.

If the relative is able to monitor the patient, CoAP GET requests are sent every minute to the CoAP remote server in order to retrieve the values.

Our OntoSmart system provides the discovery of any used BLE sensor due to the use of BLE Scanner which discovers the node, the node wrapper which adds the node's semantic description to the semantic registry, and the SPARQL agent that executes the node and service discovery queries. Moreover, publishing the monitoring patient's data through a CoAP server permits the relatives to instantly monitor these data by issuing CoAP GET requests. Further analysis and advanced diagnostics can be conducted on the remote server by domain experts in order to help the patient. Moreover, due to the use of semantic registry on the remote server, new domain expert ontologies can be imported to infer severe situations related, for example, to heredity problems or environmental diseases.

5.3. Testing & Results

In fact our testing process focused on two main principles: the requirements (as highlighted in Section I) of IoT systems and the mobile application itself. For that purposes, the applications were tested on Samsung GALAXY S4 (Android 4.4.2 KITKAT-API 19 and Android 5.0.1-API 21 Lollipop after upgrade) and SONY XPERIA SL (Android 4.1.2 Jelly Bean- API 16). We first considered one patient per WSN. We installed the application on two different patients' mobile. For each patient, one relative was added. We first insure that the ontology is well classified. Due to the use of incremental reasoner, the services were discovered sequentially. For example, once the H7 Polar is paired, the heart rate service was added on the server. Moreover, the user was able to enable/disable the services by his/her own without the intervention of experts. At certain stage, the Patient 1 decided to disable the indoor localization services. Immediately, on the relative smart phone, the location field was replaced by "Unknown". This proves the ability of our proposed system to offer services based on the occupants needs and requirements.

Then, we considered the scenario where two patients are sharing the same WSN, i.e. sharing the same Rad Beacons for the indoor localization. The distance between rooms was about 20 m and three Rad beacons were carried out and setup in three different zones at the patient's home. For testing purposes only, the remote server and the local server were on the same machine, but working on different port numbers. The ontology was well classified without any ambiguity. This insures the reusability of the proposed system by more than one occupant.

The privacy concerns were addressed by the use of email accounts for application installation and the authentication mechanism when a relative requests to monitor a patient. It is evident that this authentication mechanisms, as well as security concerns, could be enhanced by implementing advanced protocols. However, this enhancement does not affect the overall architecture, but only require the adoption of a new security/authentication agent invoked when needed.

After insuring the well functionality of the system, we focused on measuring the complexity of the system on the server. For both cases, the tests were conducted for about 8 hours. Table 2 depicts the time needed to load and classify the ontology, as well as the CPU and memory usage.

Table 2- System Evaluation on the Server

		Case 1	**Case 2**
Ontology	Load	24 sec	35 sec
	Classify	90 sec	75 sec
	Model Size	1403	1403
	Inferred Model size	1019	1000
CPU	Max. CPU	76%	60%
	Avg. CPU	35%	27%
	Memory	425 692 K	415 728 K

We can note that the ontology is classified within 2 minutes. The size of the inferred model reflects the importance of SWRL rules used in the proposed system. As few data is required to be sent for the mobile application to the semantic server, lowest bandwidth will be consumed. The potential of the semantic rules is inferring all needed data to discover the properties of a service. For example, all HR services will be inferred by just sending that the heart process is offered by H7 Polar sensor (see Rule 1 and Rule 2). We are actually working on studying the performance of the server on large scale (more than 10 patients registered in the application) to estimate the probability of crashing.

We passed than to evaluate the performance of the patient's mobile application. The time needed for node/service discovery was maximum 1 second. The patient's application was tested for about 8 hours; the battery consumption was maximum 12 % and minimum 5%, while the average CPU usage was 8%. Table 2 depicts the processing, network and battery usage on the patient smart phone in initiation mode (done only when the application is loaded for the first time) and the normal mode where the measurements are taken periodically.

This evaluation ensures that our proposed application is not dramatically affecting the resources on the smart phone. In fact, these consumptions can be decreased by the use of ambient sensors or biomedical accelerometer sensors instead of the built-in sensors. In that way, the data will be received directly on the server for semantic annotation and treatments. Aiming to determine the falling detection, we tried to test to detect a falling after standing, from bed or from a chair. These fallings were successfully detected. We can note that the posture and the falling status are not 100% accurate because we were using the Smartphone embedded sensors for the measurements (for economy purposes). These measurements can be enhanced by using dedicated accelerometer sensors capable of detecting more precisely the posture of the patient.

It is now the time to evaluate the relatives application. We first tested the authentication process. Only authenticated relatives were able to monitor the patients and receive notifications. We tested the network and battery consumption (depicted in Table 3) of the relatives for about 8 hours.

Table 3- Resources Consumption for the Patient Smart Phone

	Initialization Phase		**Main Activity**	
	Max	Average	Max	Average
CPU %	5%	2%	11%	8%
Network Usage	16 Kb	12 Kb	40 Kb	36 Kb
Battery Usage	7%	5%	12%	10%

On the relatives' side, the measured values and patient's status and alarms were displayed in real time. The CPU usage on the relative's application attempts as average 0.56 % and as maximum 1 %. These resources consumption are considered insignificant due the use of CoAP messages (1280 bytes). The values were displayed in real time and the notification when abnormal values are detected was received within 1 second.

Table 4- Resources Consumption on the Relative Mobile Application

Average CPU %	0.56%
Average Battery Usage	0.40%

6. CONCLUSION

In summary, although our OntoSmart system does not actually provide the best solution for fall detection (only because Smartphone embedded sensors were used instead of dedicated sensors), it offers the flexibility to enhance this detection, whenever needed, by just adding the necessary sensors. Our OntoSmart system is the first step toward a Sensing as Service where users can easily add sensors and actuators. Service developers will only focus on ads-on services and applications without the need to deal with sensor integration and configuration. In addition to the passive intervention devices, active intervention devices for reminder systems or medical assistance can be added to OntoSmart System. This system proved the efficiency in solving the challenges in IoT system (see Table 1). Due to the use of Service scanner and MyOntoService ontology, the users just ask for a service and it will be automatically discovered and served. The data scanner and node wrappers permit the discovery of any node regardless the physical interface. Multi agent usage, in combination with the proposed modular ontology allow to distribute the processing in different IoT nodes taking into the consideration the node's capabilities (processing, memory, battery lifetime). The use of semantic rules and SPARQL agent enhance intelligent decision making by allowing the expert to define domain based rules.

It is worthy to note that, as aforementioned, security mechanisms are not within the scope of this study. That's why we based our test-bed authentication mechanisms on Android GCM due to its implementation simplicity, and by assuming that all our system users have a Gmail account. Stronger authentication mechanisms will obviously have to be designed and implemented for a commercial version of our proposed framework. More broadly, security and privacy by design mechanisms will have to be addressed, specified and carried out within our proposed OntoSmart system for in particular critical use cases handling and personal data privacy protection. Nevertheless, we have already investigated few potential security/trust/privacy solutions that could probably be carried out in our IoT compliant and semantic based system (OntoSmart) for that purposes. These envisioned potential solutions are summarized below. It is mostly probable that, for critical e-Health uses cases, a private IoT solution would be preferred to a public one.

For security, the raw data exchanged within the patient's WSN clusters could be encrypted at the sensor/actuator level prior to its capture/relay. This could be achieved by e.g. carrying out a simplified distributed key management infrastructure, provided with distributed Certifications Authorities, within our OntoSmart system. These distributed certifications authorities could e.g. reside both in the distributed control and monitoring servers of the patient-associated caregivers and in the local servers of the patient (e.g. the local semantic server and/or the hub presented in Figure 4). The simplified keys and certificates provided within such architecture would be used within OntoSmart for: exchanged data encryption, communication channel security and authentication purposes (patients and nodes). Furthermore, the keys could also be used by the Semantic Wrapper Agent of OntoSmart '(see Figure 4) to:

•Decrypt the raw data received from Data Scanner Agents,

•Encrypt the corresponding semantic information based on our OntoSmart system ontologies.

For hiding the identity of the users, all the IDs used inside the data/ontologies would probably have to be replaced by Virtual IDs (VIDs). Only authorized entities (e.g. the patients, their relatives and their authorized caregivers) would be able (have the right) to link a VID with its corresponding real ID.

Our OntoSmart system is fully based on MyOntoSens ontology that already embeds a 'trust level' attribute within its 'Node' building block. This attribute could be in particular associated to a trust label that a certification authority could delivered based on some level of security properties/mechanisms carried out within the node. This trust label could be verifiable through e.g. a certified web-based distributed repository.

Concerning data access control and privacy, mechanisms based on data integrated privacy related attributes and rules (by design within the semantic data model and ontology) associated with semantic data annotation agents (i.e. data tattooing like agents) and rule engine agents (see Figure 4), could be envisioned. Indeed, our OntoSmart system is fully based on MyOntoSens ontology that already embeds the following attributes within its ‘Proccess and Measurements’ building block:

•Data ownership and data owner(s) ID(s) (or VID(s)),

•Data access rights and authorized data reader/writer ID(s) (or VID(s)).

In that way, the data concentrators (e.g. the semantic local/remote servers and the hubs in our OntoSmart system, see Figure 4) could decide to relay or not the information based on real time semantic annotation of data, data embedded privacy-related rules that could be added to the MyOntoSens WSN ontology, and destination node thrust level.

More agents are under construction for example, remind patients to take his/her medication or alert him/her if a stove burner is left on. Surely, actuators can be added to the system. All these concerns do not need to re-invent the wheel, but just upgrade the system by adding adequate agents. More tests are being conducted to measure the efficiency of the remote semantic storage and management server when dealing with large amount of TSHCSs. Additional scanners and writers are also under development, especially ZigBee and HL7 scanners/writers. Finally, more security and privacy concerns are under investigations.

REFERENCES

[1] C. Levy and D. Wong, “Towards a smart society,” no. June, 2014.

[2] C. Devices and F. D. Rates, “5G : The Internet for Everyone and Everything.”

[3] Huawei, “5G : A Technology Vision,” Huawei, White paer, pp. 1–16, 2014.

[4] “Smart Society project,” 2016. [Online]. Available: http://www.smart-society-project.eu/about/.

[5] A. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a Service and Big Data,” Proc. Int. Conf. Adv. Cloud Comput., no. July, pp. 21–29, 2012.

[6] X. Sheng, X. Xiao, J. Tang, and G. Xue, “Electrical Engineering and Computer Science Sensing as a service : A cloud computing system for mobile phone sensing Sensing as a Service : A Cloud Computing System for Mobile Phone Sensing,” 2012.

[7] Matt Turck, “Internet of Things: Are We There Yet? (The 2016 IoT Landscape),” 2016. [Online]. Available: http://mattturck.com/2016/03/28/2016-iot-landscape/.

[8] S. C. Workshop, “IERC - IoT European Research Cluster -Role □ Bring together the EU-funded projects and policy activities with the aim of : Sustaining Europe ’ s leading position in the future Internet of Things within a global context,” no. September 2014, 2017.

[9] T. Instruments, “The Internet of Things : Opportunities & Challenges,” p. 17.

[10] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Futur. Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, 2013.

[11] S. De, P. Barnaghi, M. Bauer, and S. Meissner, “Service modelling for the Internet of Things,” in Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, 2011, pp. 949–955.

[12] T. George and B. George, “OWL-S : Semantic Markup for Web Services.”

[13] S. N. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim, “A unified semantic knowledge base for IoT,” in Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, pp. 575–580.

[14] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, and others, “The SSN ontology of the W3C semantic sensor network incubator group,” Web Semant. Sci. Serv. Agents World Wide Web, vol. 17, pp. 25–32, 2012.

[15] V. B. M. Wick, “No Title,” 2016. [Online]. Available: http://www.geonames.org/ontology/documentation.html.

[16] W. Wang, S. De, R. Toenjes, E. Reetz, and K. Moessner, “A comprehensive ontology for knowledge representation in the internet of things,” in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, 2012, pp. 1793–1798.

[17] L. Nachabe, M. Girod-Genet, and B. ElHassan, “Unified Data Model for Wireless Sensor Network MyOntoSens Ontology,” IEEE Sens. J., 2015.

[18] L. Nachabe, B. Elhassan, J. Khawaja, and H. Salloum, “Semantic Smart Home System : OntoSmart to monitor and Assist habitant,” vol. 10, pp. 78–86, 2016.

[19] B. Parsia and E. Sirin, “Pellet: An owl dl reasoner,” in Third International Semantic Web Conference-Poster, 2004, vol. 18.

[20] J. Ferber, Multi-agent systems: an introduction to distributed artificial intelligence, vol. 1. Addison-Wesley Reading, 1999.

[21] B. Technologies, “Bluetooth ® low energy technology.”

[22] M. Lanthaler and C. Gütl, “On using JSON-LD to create evolvable RESTful services,” in Proceedings of the Third International Workshop on RESTful Design, 2012, pp. 25–32.

[23] Z. Shelby, K. Hartke, and C. Bormann, “RFC 7252: The Constrained Application Protocol (CoAP).” p. 112, 2014.

Authors

Lina Nachabe received her engineering Diploma in 2007 from the faculty of engineering of the Lebanese university, Tripoli Lebanon. She got her PHD degree from the University of Télécom SudParis, RS2M department in 2015. From 2009 she started to teach database and networking courses in Lebanese universities. She is actually a Telecom Lab Assistant in the Lebanese university, faculty of engineering. She supervised engineering project in MUT University. Lina Nachabe contributed in the ETSI technical committee TC SmartBAN, Work Item 1 "Heterogeneity management, data representation and transfer" as part of her PHD study. She is interested in wireless sensor networks and ontology researches.

Associate Professor Dr. Marc Girod-Genet received his engineering degree with distinction (top in one's year) in ‘telecommunications and advanced techniques’ from EPITA engineering school, Paris, France, in September 1994. He received the MS degree in Computer Science from Stevens Institute of Technology, Hoboken, New Jersey, USA, in Mai 1995. He finally received a Ph.D. degree with highest honors in Computer Science from University of Versailles/Paris VI, France, in July 2000. He joined Télécom SudParis first as a research project manager in September 2000 and then was confirmed as a full Associate Professor in January 2005. He teaches courses in personal communications, short range wireless technologies, service and context management architectures, optimization/modeling and machine learning. He is also an associate research at CNRS (network intelligence field) and works / has worked in several national and European research projects. Dr. Marc Girod-Genet is involved in the ETSI technical committee TC SmartBAN where he serves as Reporter of Work Item 1 "Heterogeneity management, data representation and transfer". He is /

has been a Technical Program Committee member of IEEE conferences, as well as a reviewer for conferences and journals such as Communications Magazine and Transactions on Wireless Communications. He has been involved in organizing international conferences symposia and events such as Globecom Wireless Communication symposium and ASWN workshop. In 2010, Dr. Marc Girod-Genet received 2 awards: the ITEA2 Achievement Award 2010 Silver Medal for the CAM4Home EU project and the ADEME Special Award 'Prix de la Croissance Verte Numérique' (digital green growth) for his joint work on Smart Grids with two of his colleagues.

Professor Dr. Bachar A. ElHassan got his engineering Diploma in 1991 from the faculty of engineering of the Lebanese university, Tripoli Lebanon, his MS in signal processing in 1992 from the National polytechnic institute (INPG) of Grenoble, France and his PhD in electronics in 1995 from the INPG-France in collaboration with France Telecom. Since 1996 till now he is working at the faculty of engineering of the Lebanese university Tripoli Lebanon where he is now an associate professor. He served as chairman of the electrical engineering department for six years. He also founded the Telecommunications and Networking research team at the laboratory of Electronics Systems, Telecommunication and Networking (LaSTRe) of the Doctoral School of Sciences and Technologies (EDST) of the Lebanese University. His research interest's concern digital communication and wireless sensor networks. He is the chair of IEEE ComSoc Lebanon chapter since Jan 2013.