Semantic Structure for describing IoT System: IoTContextOnt. Application for smart home and smart airport

Chaima Bejaoui a,*, Fatma Achour b and Anis Jdidi b

a LETI (Laboratory of Electronics and Information Technology), University of Sfax, Sfax, Tunisia
b MIRACL (Multimedia, Information systems and Advanced Computing Laboratory), University of Sfax, Sfax, Tunisia

Abstract. Internet of Things (IoT) is the interconnection between the Internet and objects, places and physical environments. These objects can exchange information and digital data between the real world and the Internet. The ubiquitous connectivity provided by the IoT requires integration between heterogeneous sensors, actuators, mobile phones, etc. Extending the IoT with Web Services technology is a means to achieve interoperable communications between objects and makes it easy to integrate dynamic distributed processes. The best solution seems to be the association of semantics with these heterogeneous data through ontologies in order to facilitate their reuse and allow the implementation of reasoning mechanisms. Several web ontologies, such as Open-IoT, Saref, IoT-ontology, etc., are devoted to that purpose. Since these ontologies are difficult to implement and to instantiate them in different contextual situations, in this work, we propose an ontology that ensures the description of the IoT system. We also aim to have a conceptual adaptation and make this ontology extendable core-domain.

Keywords: IoT, Design, Adaptation, Semantic, Web Service

1. Introduction

The Internet of Things (IoT) is considered as the third evolution of the internet web 3.0 following the social web or web 2.0. It represents the extension of the internet to objects or places in the physical world. It designates the connection of these objects to a wider network either through smartphones or autonomously. The IoT has evolved at an exceptional speed: connecting a large number of heterogeneous objects (sensors, actuators, smartphones, applications, etc.) [1]. IoT is defined as "an interconnection of sensors and actuators to share information across platforms via a unified framework, developing a common operational image to enable the development of innovative applications" [2]. This new type of network offers a wide range of potential applications in the fields of smart city, agriculture, home automation, smart factory, telehealth, etc. [3]. Indeed, all these connected physical objects will be able to provide new services through sensors (temperature, presence, pollution, etc.) and actuators (remotely controllable devices acting on the environment) and other entities. Furthermore, they can be considered in themselves as new services [4] in a vision of service-oriented computing (SOC). Nowadays, the application domains in IoT are vast and the challenges are still numerous: both syntactic and semantic interoperability problems, the complexity of managing these complex systems and integrating them with other systems as well as already existing services [5]. Interoperability is still a relevant issue. Indeed, this vast and complex ecosystem of technologies, protocols

*Corresponding author. E-mail: bejaouichaima@gmail.com.
and standards makes the deployment of applications integrating different technologies or different sources of data complex. A necessary first step is syntactic interoperability: to be able to use different technologies seamlessly and homogeneously by integrating heterogeneous and incompatible technologies. Several Web ontologies and description frameworks have been dedicated to this goal. The semantic ontology of the IoT must allow a complete understanding of the flows, services and qualities of the system. IoT ontologies and reference models tend to focus on defining the entities, their properties and the controls involved in the IoT network (for example, system structures) [6], (2017). Several ontologies have been proposed to deal with different aspects of IoT-based sensor data collection ranging from the discovery of IoT sensors for data collection to the application of reasoning on the sensor data collected to draw inferences. Generic and other specific ontologies exist in the literatures. In this paper, we are interested in the IoT ontology, namely generic ontology. This work aims to ensure the conceptual adaptation of the IoT system, facilitate the merging of data collected in different domains for horizontal applications and allow the ontology to be an extendable core-domain. Our proposition called "IoTContextOnt" regroups essential information about the IoT system. This remainder of the article is organized as follows: Section 1 presents the definition of IoT, its architecture, application areas and major challenges. Section 2 gives a brief state of the art of IoT ontologies, which are related to our objective in the present work. Section 3 describes the proposed "IoTContextOnt" ontology. Section 4 explains our adaptation proposed in the IoT system by a general architecture for the adaptation of semantic data and using a smart home and airport as a case study. Finally, Section 5 concludes the paper and presents future work.

2. Internet of Things

There are multiple definitions for IoT but they all have the same concept: The Internet of Things (IoT) is a network of networks that allows, via standardized and unified electronic identification systems and mobile wireless devices, to directly and unambiguously identify digital entities and physical objects and thus to recover, store, transfer and process, without discontinuity between the physical and virtual worlds, related data. This combination connects the physical world and the virtual space via the intelligent device, thus, improving the connectivity of any object at any time in any location to solve any problem [7]. According to [8], "The Internet of Things (IoT) is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals”. It is also defined as an open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, react and act in face of situations and changes in the environment [9]. We define IoT as a set of entities called sensors that detect environmental events and transmit them to actuators that modify the behavior or a state of a system. The collected information are transferred to users via the Internet to provide them with services of daily living. Generally, these data are heterogeneous because they come from several heterogeneous nodes.

2.1. Architecture

The architecture of an IoT system is composed of several levels that communicate with each other to connect the physical world of objects to the virtual world of networks and the cloud (see Figure 2). All the projects do not adopt a formally identical architecture; nevertheless, it is possible to schematize the course of the data [10]. In this paper, we use the basic architecture of IoT[11] which consists of five layers.

1) The perception layer usually consists of different types of sensors. Its essential role is the identification and collection of specific information by each type of sensor device. 2) The network layer plays an important role in routing sensitive information from sensors to the processing system. Therefore, this layer is mainly responsible for the transfer of information from the perception layer to the upper layer. 3) The intermediate layer has two main functions: service management and storage of information from the bottom layer in the database. 4) The application layer is responsible for managing applications based on information processing in the middle tier. 5) The management layer covers the management of applications and services of an intelligent system. It can create charts, flow charts, a report, and so on based on the data received from the lower layer and the efficient data analysis processes.
2.2. IoT applications domains

Although few years ago, connected objects have been unknown topic: they are now everywhere as they invaded our daily lives: from smart TV to connected car influencing our hobbies and travels and increasing our comfort. The potential of connected objects is enormous[12]. A recent 2016 study has predicted that more than half of the tools and trades will make calls to IoT in 2020. The applications are varied and cover many areas: industry, science, health, agriculture (An IoT-based cargo monitoring system for enhancing operational effectiveness under a cold chain environment)[13], etc. In what follows, outline some examples of IoT application fields.

- The medical domain: The standardization of the IoT in the field of Health[10], [14], [15] has contributed to the opening of new horizons as they assure surveillance in medical institutions and maintenance, surgical operations and remote control, geolocation services, medical diagnostics, etc. We can cite an example in the medical domain [15]. In this work, The authors proposed an architecture called IoHT-CLS that provides different technologies for the management of healthcare facilities for the elderly. The objective is to collect data and transfer it via an adaptive neurofuzzy inference system for elderly care facilities. This architecture combines a structured framework on the integration of internet of healthcare things (IoHT) and artificial intelligence.

- The transportation and logistics domain: Applications in the field of the automotive industry [16] are numerous and require the use of "smart objects". These can monitor and report several things from tire pressure to proximity to other vehicles. The connected vehicle allows car manufacturers and suppliers to develop new services. For example, remote geolocation and recovery of stolen vehicles. In [17] the authors proposed an IoT-CMS, integrating both IoT concepts and AI techniques to monitor any environmental changes of environmentally sensitive products in order to ensure their functional quality throughout the entire cold chain operational environment. The proposed system architecture consists of three modules; the data acquisition module (DAM), the storage condition adjustment module (SCAM), and the guidance establishment module (GEM).

- The domestic domain: Defined by the set of techniques aiming to integrate all automation in the field of security, energy management and communication into the home. Home automation is a relatively new practice, but one that is constantly evolving [16]. For example, control and programming of various interventions inside the home, information capture (alarm system, temperature variations, weather forecast, etc.), control actuators which allow the programming and control of the various electronic devices at home even remotely.

- The energy domain: The digital revolution has entered the energy sector through resource management. In the context of energy, it answers major issues: depletion of natural resources, increasing global energy needs and lack of human labor. The European Union officially adopted a 20-20-20 renewable energy directive setting climate change reduction target for the next decade [5]. Including the objective of having a good gas reduction, a better energy efficiency and a 20% increase in the use of renewable energies by 2020. The examples of the applications described above show that the IoT is integrated into our daily life and contributes in improving its quality.
Agriculture domain: In the agricultural world, IoT solutions take the form of sensors connected to the Internet to collect environmental and mechanical measurements [18]. Their deployment allows farmers to make informed decisions and improves almost every aspect of their work, from farming to farming. The farmer has to feed the data about the land area and the type of crop planted, also provides the information collected from sensors about the soil fertility, humidity, water overflow, field animals.

2.3. IoT challenges

All information (textual, audio or video, but also gestures are in permanent interaction with all the other types) from the IoT system become reticular. IoT transforms the Internet into a globalized hyper-network with all the points forming as many nodes of information as possible. Indeed, a mass of heterogeneous information shared and transmitted can generate a list of important challenges: heterogeneity, scalability, interoperability, security, extensibility management and energy efficiency [19], [20], [21]. Heterogeneity, scalability, and interoperability will be of primary importance in a complex and dynamic system like IoT.

- **Heterogeneity**: The IoT is directly affected by the diversity of hardware and software components used to build objects [22]. Indeed, depending on the manufacturers, the objects do not use the same operating systems, do not have the same communication interfaces and do not respect the same data format, which leads to a significant technical heterogeneity. Moreover, the formats used to name objects, to describe their identities and to represent the data their sensors produce are numerous, which makes it difficult for two objects to exchange "understandable" information.

- **Scalability**: IoT allows you to interconnect a large number of different devices in a single application and also allows having different applications in the same place. The devices and applications as well as the services offered must be extensible. In addition, evolution in terms of data is very important. Hence, the need to choose effective ontologies and semantic methods [19].

- **Interoperability**: Interoperability is a computer term for systems capable of adapting and collaborating with other independent systems that already exist or that will be created. This compatibility makes it easy to create a network and transfer data from different programs. The interoperability between heterogeneous devices and their models in IoT should be considered. Preferably, it should also be able to analyze data from other types of operating software in order to reduce the investments as much as possible.

- **Security**: Security in IoT systems and applications is a very important issue. These applications require data confidentiality, authenticity, integrity, access control, authorization, non-repudiation, availability and privacy [19], [23] within the IoT network. Once the security objectives are determined, the risks weighing on each of these elements can be estimated according to the threats. The overarching goal is to prevent unauthorized use, misuse, modification or misuse of information circulating on the IoT system.

- **Management**: Management is another challenge for IoT. It englobes the devices management, data management, services management, protocols management, etc. [19], [12], [20], [21], [24]. The service management platforms are able to integrate different infrastructures and service components according to specific application domain requirements. For example, Light-weight M2M (LWM2M) [16], [25] is a standard that has been developed by the Open Mobile Alliance to provide interface between machine to machine (M2M) devices and M2M Servers to build an application agnostic scheme for the management of a variety of devices.

- **Energy efficiency**: Energy consumption is one of the most difficult aspects in the design of IoT systems. Generally, the sensors are equipped with batteries that are unchangeable and thus we need to increase the system’s lifetime. Among the solutions for power collection is cloud computing [26] which allows us to make the mobile devices resourceful in terms of computational power, memory, storage, energy and context awareness.

- **Extensibility**: The protocol should adapt to a network having a size that ranges from 100 to 1,000,000 nodes both in terms of performance and memory usage: an increase in the size of the network may not lead to an increase of the size of the routing table Truong (2014). In what follows we will focus on data management in contextual and semantic terms.
3. Related work

The fields of application in IoT are vast and the challenges are still numerous today: both regarding the problems of syntactic or semantic interoperability as well as the complexity of managing these complex systems and integrating them with already existing systems and services. A necessary first step is syntactic interoperability: being able to use different technologies in a transparent and homogeneous way by integrating heterogeneous and incompatible technologies. An additional step lies in semantic interoperability: being able to use different technologies in a harmonized way is one thing; however, allowing systems to automatically understand the different data produced by these heterogeneous sensors with disparate data formalisms remains a challenge [5].

Research initiatives and standardization activities in areas allied to the IoT vision have mainly focused on sensor descriptions and observation data modeling. An IoT ontology is important to represent Internet of Things (IoT) resources, entities and services. Thus, the core modeling concept considered in IoT ontology is "resource", with all sensors, actuators, and processors being modeled as resources [15]. A resource model captures resource functionalities, and where and how they can be accessed, in a conceptual view. The concrete instantiation of this information is contained in the resource description, which is published in a resource directory that acts as a service repository.

The syntax and semantics of the interfaces are captured in the advanced resource description, which is an ontology including concepts such as location, type (Sensor, Processor, Actuator), and operations of a resource. Thus, more modeling concepts are needed to link the sensor descriptions to sensor measurements and then to the observed entity in the IoT domain. The entity constitutes "things" in the Internet of Things and could be a human, animal, car, store or logistic chain item, electronic appliance or a closed or open environment. The "entity" is the main focus of interactions by humans and/or software agents. This interaction is made possible by a hardware component, called "device", which either attaches to an entity or is part of the environment of an entity so it can monitor it. The device allows the entity to be part of the digital world by mediating the interactions. The actual software component that provides information on the entity or enables controlling of the device, is a "service". As implementations of resources can be highly dependent on the underlying hardware of the device, a "service" provides a well-defined and standardized interface, offering all necessary functionalities for interacting with entities and related processes.

To overcome this problem, several ontologies have been studied by researchers to describe IoT systems and foster interoperability. These ontologies are classified into generic and specific and are characterized as follows [27], [28]:

- IoT ontology should include concepts to define the context of the data source (for example, sensor mobility, current activity, if the measurement was taken automatically by the device or in the event of human intervention).
- The IoT ontology must integrate the different ways of locating the data source to allow developers to identify the data source.
- An IoT ontology should provide concepts to support different formats for defining the time of data collection.

These ontologies meet specific needs in the field of IoT. In what follows, we will present a state of the art of ontologies in the IoT-domain.
3.1. Generic ontologies

IoT-O is one of the generic ontologies, which represent a modular IoT ontology of the main domain offering a vocabulary describing the connected peripherals and their relationship with their environment. Its initial structure was proposed [28], and an enriched version, structured by design patterns was presented in [29]. In fact, IoT-O is a unifying ontology and brings together many concepts defined in recognized ontologies. The design of IoT-O follows the NeON methodology [30]. This ontology includes concepts of both connected objects (actuator, sensor) and various metadata on the physical objects, but also links with services related to these physical objects. It provides classes and relationships to link modules such as sensing module, acting module, life cycle module, service module, and energy module. IoT-O’s core is limited: it defines 14 classes, 18 object properties, and 4 data properties. IoT-O reuses the concepts of SSN, SAN, DUL, QUDT, one M2M and more, defines new concepts.

Semantic Sensor Network (SSN) [31], [27] ontology describes the sensor resources, the data collected through these sensors, and observations. SSN ontology seeks to resolve the heterogeneity problems that are associated with sensor discovery and sensor data collection, but its concepts are limited to support the spatial and temporal association of sensor data with resources. It defines 10 classes and 16 properties. The full ontology consists of 41 concepts and 39 object properties. The SSN also supports domain ontology extension which is M3 ontology [32]. OpenIoT [33] ontology describes abstraction of sensors and their integration with cloud computing concepts. OpenIoT bases on the SSN ontology and the sensor data, which reuse the concepts of Observation, sensor, location and test beds. It further extends to define utility metrics for service level agreements between its users and the OpenIoT services provided. OpenIoT provides an open interface for integrating heterogeneous sensor data into a common information model. The main purpose of this ontology is to provide a semantic description, which enriches the sensor data and the context. In addition, it provides several services to structure reasoning queries in order to analyze the data. In [34], the authors consider the efficient way to find and construct a workable model to access and extract application project knowledge is to exploit ontology as a design pattern. The proposed ontology in IoT application project is divided into 3 layers: a physical layer, as things; an information layer, as data and metadata about knowledge provided by things; and a functional layer comprising services provided by things. The Device Ontology models actual hardware devices that may exist in the network. The domain Ontology models information about real world physical concepts and their relations among each other. Finally the estimation Ontology contains information about different estimation models, the equations that drive them, the services that implement them, and so on. [35] propose a semantic sensor network integration technique which is used with the ontology alignment extraction technology. This work aim to improve the quality of ontology alignment by using correctness factor of each matcher to determine the correspondent’s global factor, and also using the strength of the support strength and disprove strength in the debating process to calculate its local factor. The sensor ontology extraction approach aims to find high-quality alignment from various sensor ontology alignments, which can be used to bridge the semantic gap between heterogeneous sensor ontologies and integrate the knowledge defined inside.

3.2. Specific ontologies

The Smart Appliance REFerence (SAREF) [36], [37] is a unified ontology dedicated to energy management and services in smart homes. SAREF facilitates the mapping of existing assets (standards / protocols / models / data etc.) in the field of smart devices. The SAREF ontology provides building blocks that allow the separation and recombination of different parts of the ontology according to specific needs. The concept of sensor is redefined, while it is present in SSN, also it covers actuators. The concept of functions (one or more commands supported by sensors, which are defined in SAREF) supports the modularity and extensibility of ontology, and contributes to the maintenance of equipment. This is a new smart home ontology among the 45 smart home ontologies [32] already existing. The ontology has been extended [37] to apply to other areas of IoT such as connected buildings, etc. The IoT-Lite [19] is lightweight core domain ontology. High-level concepts are defined as actionable object or services. It has been adopted in various projects as a core model such as FiWARE for example due to its simplicity. IoT-Lite provides a lightweight ontology instantiation of SSN and extends it with SAO. In fact, IoT-Lite includes dynamic semantics to estimate missing values from sensors during the data annotation phase. This saves time, since it is no longer necessary to send the data to a server to extrapolate the missing values from the sensors. This time...
advantage makes the IoT-Lite dynamic, fast and interoperable. Concepts covered by IoT-Lite include information
and sensor locations.

[38] propose a model of ontology that is specific to a domain called University activity ontology (UAO) in 2017.
This model is designed based on the indoor activities at a university in order to have an efficient management of
semantic data and requests. The authors use a hierarchical structure by decomposing their proposal into two sub-
ontologies. A first called "Upper level" which reuses the concepts of context ontology Wang et al., (2004). A second
called "lower level" which is a domainspecific ontology. The main advantage of this division is the improvement in
reusability and the readability of this ontology.

The authors in [39] propose a contribution that is based on a system for the early detection and monitoring of
COVID-19 based on an ontology method using 1D sensory biomedical signals such as ECG, PPG, temperature
and accelerometer. The aim is to provide information about the corona virus patient in any location. For this, the
authors use three technologies such as RFID, microcontrollers and sensors. In addition to using the ontology related
to covid-19.

To summarize, existing published IoT ontologies are either generic or specific-domain. The studied ontologies
propose redefinitions of the service concept and become reused ontologies. Such as IoT-O, SAREF, OpenIoT, and
IoTLite. The reusability of all the strong concepts of different standard ontologies or already recognized in the
domain is a strength for the ontology and it is therefore interesting to be able to be based on such an ontology of
unifying domain. The main objective is to define shareable and understandable vocabularies for IoT systems. Indeed,
these ontologies promote interoperability between objects, and make data usable automatically. In this sense, we
propose our new ontology which will be explained in detail in the following section.

4. The proposed adaptation in IoT system

4.1. "The "IoTContextOnt" Ontology

In this paper, we present an ontology used to ensure the description of the IoT system. This ontology is the result
of the three ontologies, namely Open-IoT, Saref and IoTontology, which present the set of the generic ontology
used in IoT system. The latter is a OWL Ontology representing the intersection result of three existing models to
describe the IoT System: Open-IoT, Saref and IoT-ontology. The goal is to ensure the conceptual adaptation of the
IoT system. In fact, to be reusable in a wide scope of domains, our IoT ontology contains a set of key concepts. These
are representative of IoT systems with no regard to the application domain. This approach facilitates the merging
of data collected in different domains for horizontal applications and allows the ontology to be an extendable core-
domain. The main objective is to define shareable and understandable vocabularies for IoT systems. Indeed,
these ontologies promote interoperability between objects, and make data usable automatically. In this sense, we
propose our new ontology which will be explained in detail in the following section.

An IoT system regroups several types of devices, sensors and actuators. Two types of devices exist "sensing Device"
and "Actuating Device". However, "Situation" of IoT context comprises "tasks" and "services" with a specific
"location". An IoT system presents various types of property as "Time", "Energy" and "latency". In addition to
the environmental data, we also store the information of location members, their preferences and activities as well.
Each ontology class has the relationship so that we can adapt information using one Jena toolkit rule. The set
of "IoTContextOnt" classes is not exhaustive; we can also add other classes as "Temperature" class that can be
presented as a property of IoT system.

The classes defined in this ontology (see Figure 3) are described below:

- "Device", constitutes the basic component of an IoT context, composed of both physical and virtual elements.
The devices can be of two principle types that are not mutually exclusive, i.e. "Sensing Device" and "Actuating
Device".

- "Sensor", are devices acquiring data. This class describes the acquisition context and the data collected by the
system. Sensors capture the system's perception of the evolution of its environment.

- "Actuator", is the device that enables the system to act on the physical world. Actuators capture the information
the system has about its own abilities to impact its environment and to make it evolve.

- "Situations", are the descriptions of contextual situations in IoT system. These concepts present both "task"
and "service" classes executed in IoT system.
Fig. 3. The “IoTContextOnt” ontology

- "Task", stands for the tasks executed in IoT system. The proposed ontology is generic and executed in a large variety of IoT systems. These are important to present tasks of this type of system.
- "Service", in many cases, the IoT and the programmable web are very close. Connected devices can be seen as service providers and consumers. By specifying a notion of service, every aspect of an IoT system can be represented.
- "Person", presents the existing users in IoT system. This class describes the contextual information about the users’ profiles. There are two principal types of users: "user" and "client".
- "User", is the group of existing users in more or less permanent way in smart location.
- "Client", is the invited user in the smart location described in IoT system.
- "Activities", are the activities of persons existing in IoT system. It presents data about IoT system members and presents three data types to answer questions about when, who and what the person does.
- "Preferences", are the preferences of persons existing in IoT system. It presents data about IoT system members’ preferences.
- "Location", presents the location where an IoT system is executed. In this type of system, we only talk about the smart places as the smart home.
- "Network", represents the class of networks related objects. This class describes the used network to connect objects in the system. It concerns the network constraints as bandwidth, capacity and scalability.
- "Energy", in the paradigm of ambient computing, many distributed objects perform computations. Most of these objects being physical devices, a complete modeling of the system will include a description of their energy consumption. Energy management is a crucial topic in IoT systems.
- "Property", any IoT system presents properties to ensure the system execution. This class regroups the "time", "latency" and "energy" characteristics. Other properties can be added such as temperature
- "Time", it is very important to present the time of capture and the perception the system to impact this environment.
- "Latency", is related to the time of system or network response.

The “IoTContextOnt” regroups essential information about the IoT system. This ontology is used to design an IoT system as smart home or smart city to ensure conceptual adaptation. To do it an OWL-S ontology is used by semantic web service, this ontology is used to semantic design of IoT system situation. The figure 4 below illustrates the extended OWL-S ontology. The extended OWL-S ontology includes the basic information examining an IoT
Fig. 4. The OWL-S proposed structure.

system. This is presented using "IoTContextOnt" class. This structure is used to create a semantic web service to
create an RDF instance for the data semantization description.

– The "Service" ontology presents the main class of OWL-S. It covers the web service general properties. This
class has a "ServiceProfile", described by "ServiceModel" and supports a "ServiceGrounding".

– The "ServiceProfile" ontology explains the functionality of services and what it requires to other agents. It
describes the service in terms of what it does to allow a requester to see if the proposed service suits him.

– The "ServiceModel" ontology defines the functioning of the web service and servers to explain how the service
works. A service can be seen as a process. For this reason, the "ServiceModel" has a sub class "ProcessModel".

– The "ServiceGrounding" ontology provides the information necessary for the use of the web service. In fact,
you should choose the appropriate protocol to access the service, the message format, the way to realize, the
used transport mechanisms and the way the addressing mode is used.

– The "Resources" ontology provides information about the resources used by the web service. Web services
often need resource to run. These resources are numerous and various; it is therefore interesting to regroup
them in one ontology.

– The "IoTContextOnt" ontology is added to the OWLS structure to create a semantic web service to regroup
the data submitted by various types of sensors in semantic instance. The "IoTContextOnt" ontology is described
using "ServiceProfile" ontology. The proposed adaptation in IoT system.

4.2. General Architecture for Data semantic adaptation

The existing IoT system is automatically adapted using the data received from sensors and actuators. The pro-
posed architecture is designed to prepare data to developers in order to develop an IoT system. Generally, sensor
data are full of diversity, ubiquity and volatility, which pose challenges for machines to understand and process.
Data semantic adaptation allows users to reason about human’s activities related to sensor events and make effec-
tive responses to dynamic environments. To this end, we proposed two principal steps; the first one regarding data
semantic description and the second conceptual adaptation. The first step result is a semantic description described
using RDF language and conform to the proposed ontology. The second step results in an adapted semantic de-
scription that is used to impact the system to its environment. As depicted in Figure 5, the processing architecture
is composed of four parts. Part 1 is a physical layer, which consists of various sensors responsible for sensing data
from ambient environments. Part 2 is the core component named data semantization description, which is composed
of three steps, namely data collection, data reprocessing and semantic description. The first two steps are regarded
as preliminary work of data semantization. Part 3 is an adaptation layer, which is composed of semantic data result-
ing from step 2 and semantic rules defined automatically according to the user’s needs. Part 4 is the final semantic
description format after adding semantics to sensor data and applying semantic rules.

– The data Collection, in this step, the main task is to sense and collect heterogeneous data from diverse sensors
including sensor id, value, measurement and other information. For separate sensor nodes, the data is submitted
to a processor via wireless communication technologies. However, for sensor networks, a main challenge is
how to arrange the roles of all sensor nodes based on the requirements and limited resources constraints as well
as the protocols used for communications between networks.
– The data preprocessing is the data collected from sensors are full of uncertainty and noise, which may result in severe problems with regard to data use. For example, in applications where data are adopted for trend predictions, the more accurate the data are, the more reasonable the predicted trend is. It is undeniable that anomalies or outliers are essential in the case of discovering abnormal situations. For instance, when the temperature is received from the sensor, the system must differentiate between human and other temperature types. However, there exist situations where noise, anomalies and outliers need to be tackled and cleaned. By adopting data preprocessing algorithms, accuracy of sensor data would be improved and it is beneficial for further processes.

– The instance description is regarded as the key step in the whole processing architecture, which means create semantic instance using the proposed ontology. The instance is created using RDF language for example. Generally, instance description is composed of two steps, semantic modeling and instance annotation. The former serves as an important role, and users may define new semantic models or reuse existing ones depending on the situations. The preprocessed data would be instantiated based on predefined ontology to finish the process of instance description.

– The semantic data is the data semantization description result, which is an RDF instance describing an IoT system context. This instance is conform to proposed “IoTContextOnt” ontology and is used to ensure the conceptual adaptation which is edited by the user to be validated.

– The semantic rules are defined automatically according to users’ needs. These rules are generics and are defined for use in different contextual situations of the IoT system. In this architecture these rules are created using Jena language. In fact, Jena framework is a free and open source of Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS, OWL, a query engine for SPARQL and it includes a rule-based inference engine.

4.3. Key Techniques

As presented in the semantic conceptual adaptation architecture (see Figure 5), the process of data semantization description includes three parts, namely data collection, data preprocessing and instance description. The semantic data adaptation includes two parts, i.e. semantic data and semantic rules. In this section, key techniques involved in each part are discussed in the data semantization description and semantic data adaptation. As shown in the conceptual data adaptation architecture (see Figure 6), the conceptual adaptation is realized by the semantic web services creation. To this end, two web services are created. The first one is created to intercept data collected from sensors, interpret these data and create a semantic structure using OWL classes and individuals. The second one is created to ensure the adaptation using semantic rules. The adaptation result is used by actuators to adapt an IoT
environment. In this environment, diverse devices, sensors and actuator are presented. The information are provided by several sensors, the semantic web service role is to provide a semantic description for these data. This description is created using the standard model for data interchange on the Web RDF, it is presented as a model that is conform to the proposed "IoTcontextOnt" ontology. The created model is used to develop a structure presenting the desired action to link the system to the physical word. In fact, this structure is used by actuator to change the system. To realize this, generic semantic rules, which are defined using Jena toolkit, are proposed to transform the initial model. Sensors are the material layer of conceptual data adaptation. They send data to Web service to ensure the semantic description. In IoT system, several sensor types exist and several data types are submitted.

Semantic web service (1) is created to ensure the data semantization description. Sensors submit raw data and the role of the semantic web service is to transform this data to semantic description. In the IoT system, several sensor types exist. As a result, it is necessary to distinguish the source of each data as well as the semantic extracted from it. Semantic web service (2) is used to ensure the conceptual adaptation. It uses the semantic description submitted from the Semantic web service (1) and the generic semantic rules to create an adapted semantic model, which is used by the developers to transform an IoT system according to users' needs.

The adapted data are used by semantic web service (2) to inform devices of the best conceptual solution in the current contextual situation. The sensors can notify the semantic web service (2) if there are new changes in the contextual situation.

4.4. Case study

Situation 1: Smart Homes are intelligent environments equipped with diverse devices, actuators and sensors. The primary goal of smart homes is to monitor inhabitants and activities in order to enhance the quality of life and enable assisted living. Researchers all over the world devote huge efforts to the development of Smart Homes. However, one inevitable challenge in Smart Homes is the heterogeneity which makes information difficult to communicate and interact with each other.

In this smart location, we present a situation that which is characterized by the presence of information about the environment and the home members, their preferences, and their activities. The properties of the location are the values of the Humidity (63%), Temperature (15 degree C), and Irradiation (4.2 kWh/m²). We note the presence of two persons: father and son at the same time "17:00:00". The father named John, his preferences are to activate the TV light, the projector, and the air conditioner during the execution of his activity. He begins his activity at "18:00:00" which consists of an interview undertaken. Furthermore, the son named Pablo, his preferences during his study (activity), are to inactivate the TV and the projector, and activate the light and the air conditioner. He started his activity at "19:30:00" (see Table 1). The realization of ontology based on our definition of IoT ontology above is relatively eligible for Smart Home. The extraction of all sorts of ontologies are shown in Figure 3. As can be seen in this figure, the ontologies are modeled in RDF/OWL language. Applying language rule is one of the most important steps in adaptation using ontologies. After building the data set, the conceptual data adaptation system
operates some rules from these created ontologies. In fact, we create generic rules that can be applied in various situations of IoT system. The RDF description relative to the first situation is presented in the Box 1. This description is in accordance with the proposed OWL ontology (see Figure 3).

In the box 1, we presented a RDF description of the first smart home situation. In this description, we used a "IoT-ContextOnt" as resource with the identifier "0012245500" and we used the concepts "Property", "Time", "Persons", "Activities", "Preferences" as properties. In each property we defined resources as "Property" which pools the resources "Humidity", "Temperature" and "Illumination" the value assigned to resource "Humidity" "20.0", the value assigned to resource "Temperature", 30.0 and the value assigned to resource "Illumination" "200.0".

Finally the preferences of each person are described by "Preferences" property that regroups the preferences of the father and of the children. The father’s preference is described by "Projector: true", "AirConditioner: true", "Light: true" and "TV: true" resources. The children’s preference is described by "Projector: true", "AirConditioner: false", "Light: true" and "TV: false" resources.

Box 1.

1: <rdf:RDF
2: xmlns:rdf="http://www.w3.org/1999/02/ 22-rdf-syntax-ns#"
3: xmlns:rdfs="http://www.w3.og/2000/ 01/rdf-schema#"
4: xmlns:xsd="http://www.w3.org/2000/ 10/ XMLSchema#"
5: xmlns:iotc="http://Ontology.IoT.com/ IoTContextOnt/0.1/">
6: <iotc:IoTContextOnt rdf:ID="_0012245500">
7: <iotc:Property rdf:parseType="Literal">
9: <iotc:Temperature rdf:datatype="http://www.w3.org/2001/ XMLSchema#double">30.0</iotc:Temperature>
11: </iotc:Property>
12: <iotc:Person>
13: <iotc:Name>John</iotc:Name>
14: <iotc:Role>Father</iotc:Role>
15: <iotc:Name>Pablo</iotc:Name>
16: <iotc:Role>Son</iotc:Role>
17: </iotc:Person>
18: <iotc:Preferences>
19: <iotc:TV>true</iotc:TV>
20: <iotc:Light>true</iotc:Light>
21: <iotc:Projector>true</iotc:Projector>
22: <iotc:AirConditioner>true</iotc:AirConditioner>
23: </iotc:Preferences>
24: <iotc:Activities>
26: <iotc:Who>Father</iotc:Who>
27: <iotc:Do>Entertain</iotc:Do>
28: </iotc:Activities>
29: <iotc:Preferences>
30: <iotc:TV>false</iotc:TV>
31: <iotc:Light>true</iotc:Light>
32: <iotc:Projector>false</iotc:Projector>
33: <iotc:AirConditioner>true</iotc:AirConditioner>
34: </iotc:Preferences>
35: <iotc:Activities>
37: <iotc:Who>Son</iotc:Who>
38: <iotc:Do>Self-study</iotc:Do>
39: </iotc:Activities>
Box 1.

Table 1
Smart Home values description of contextual situation.

<table>
<thead>
<tr>
<th>Situation 1: Smart Home</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Home Properties</strong></td>
</tr>
<tr>
<td>Humidity</td>
</tr>
<tr>
<td>20.0</td>
</tr>
<tr>
<td><strong>Time</strong></td>
</tr>
<tr>
<td>Date</td>
</tr>
<tr>
<td>20191211</td>
</tr>
<tr>
<td><strong>Person</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>John</td>
</tr>
<tr>
<td>Pablo</td>
</tr>
<tr>
<td><strong>Preferences</strong></td>
</tr>
<tr>
<td>TV</td>
</tr>
<tr>
<td>True</td>
</tr>
<tr>
<td><strong>Activities</strong></td>
</tr>
<tr>
<td>When</td>
</tr>
<tr>
<td>180000</td>
</tr>
<tr>
<td>190000</td>
</tr>
</tbody>
</table>
<iocc:Property rdf:datatype="http://www.w3.org/2001/XMLSchema#double">30.0</iocc:Temperature>

<iocc:Illumination rdf:datatype="http://www.w3.org/2001/XMLSchema#double">200.0</iocc:Illumination>


<iocc:Date rdf:datatype="http://www.w3.org/2001/XMLSchema#double">17:00:00</iocc:Date>

<iocc:Persons rdf:resource="#Father">
  <iocc:Person rdf:id="Father">
    <iocc:role rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Father</iocc:role>
    <iocc:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">John</iocc:name>
  </iocc:Person>
  <iocc:Person rdf:id="Son">
    <iocc:role rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Son</iocc:role>
    <iocc:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Pablo</iocc:name>
  </iocc:Person>
</iocc:Persons>

<iocc:Activities rdf:resource="#Entertain">
  <iocc:Activity rdf:id="ofChildren">
    <iocc:When rdf:resource="#_193000"/>
    <iocc:Who rdf:resource="#Son"/>
    <iocc:Do rdf:resource="#Self-study"/>
  </iocc:Activity>
</iocc:Activities>

<iocc:Preferences rdf:resource="#ofFather">
  <iocc:Projector rdf:resource="#true"/>
  <iocc:AirConditioner rdf:resource="#true"/>
  <iocc:Light rdf:resource="#true"/>
  <iocc:TV rdf:resource="#true"/>
</iocc:Preferences>

<iocc:Preferences rdf:resource="#ofChildren">
  <iocc:Projector rdf:resource="#true"/>
  <iocc:AirConditioner rdf:resource="#true"/>
</iocc:Preferences>
In the first rule, we define the activity of person in the input as "Entertain" and the Time as "18:00:00", the output for preference TV is "false", projector is "true", AirConditioner is "true" and Light is "true".

The first Jena rule regroups the prefix used to define the namespace of the concepts, we presented "prefixiotc" of the proposed ontology. This rule regroups the input and the output of the first situation. It regroups the concepts "iotc:who=?", "iotc:when=180000" and "iotc:do=Entertain" the existed in the "iotc:activity" property. The outputs of this rule is created by the update of the values assigned to the "iotc:Preference" property. It regroup the concepts; "iotc:TV=false", "iotc:Projector=true", "iotc:AirConditioner=true" and "iotc:Light=true". The result of the rule execution engine is to change values in the first description.

In the second Jena rule, we define the activity of person in the input as "Self-study", the output for Preferences; TV and Projector are "true" and Light and AirConditioner are false.

In the Box 3, the Jena rule regroups the prefix used to define the namespace of the concepts, we presented "prefixiotc" of the proposed ontology. This rule regroups the input and the output of the first situation but for the user when "Activity" is "Self-study". It regroups the concepts "iotc:who=?", "iotc:when=193000" and "iotc:do=Selfstudy" that existed in the "iotc:activity" property. The outputs of this rule is created by the update of the values assigned to the "iotc:Preference" property. It regroup the concepts; "iotc:TV=false", "iotc:Projector=false", "iotc:AirConditioner=false" and "iotc:Light=false".
"iotc:AirConditioner=true", "iotc:Light=true". The result of the rule execution engine is to change values in the first description.

Box 3.

1: @prefix rdf:
2: http://www.w3.org/1999/02/22-rdf-syntax-ns#
3: @prefix xsd:
4: http://www.w3.org/2001/XMLSchema#
5: @prefix rdfs:
6: http://www.w3.org/2000/01/rdf-schema#
7: @prefix iotc:
8: http://Ontology.IoT.com/IoTContextOnt/0.1/
9: [IoTModel:
10. (?d1 rdf:type iotc:IoTContextOnt),
11. (?d2 rdf:type iotc:activity),
12. (?d4 iotc:person ?person)
13. (?d2 iotc:who ?person),
14. (?d2 iotc:when '193000'^^xsd:time),
15. (?d2 iotc:do 'Self-study'^^xsd:string),
16. ->
17. (?d3 rdf:type iotc:Preferences),
18. (?d3 iotc:Projector 'true'^^xsd:boolean),
19. (?d3 iotc:AirConditioner 'true'^^xsd:boolean),
20. (?d3 iotc:TV 'true'^^xsd:boolean),
21. (?d3 iotc:Light 'true'^^xsd:boolean),
22. (?d3 iotc:TV 'false'^^xsd:boolean),
23. (?d3 iotc:Light 'false'^^xsd:boolean),
24. (?d3 iotc:Projector 'false'^^xsd:boolean),]

Situation 2: Smart Airports are intelligent environments equipped with diverse devices, actuators and sensors. We present in this location various characteristics as the location, and the number of terminals, gates and passenger. The primary goal of smart airport is to monitor inhabitants and activities in order to enhance the quality of services and enable assisted living. Researchers all over the world devote huge efforts to the development of Smart Airports. However, one inevitable challenge in Smart Airports is the heterogeneity which makes information difficult to communicate and interact with each other.

In this smart location, we present the presence of information about the environment and about a smart airport. In the airport in France, 12000 passengers exist in the different gates in 15/11/2020 at 21:00:00. In this smart location and at this time, there is a person named Thomas who waits for his flight which is after 30 minutes. He is in gate 23 and he needs to send a video footage to the company where he works. The network quality in gate 23 is bad since there are several passengers in both gates 22 and 23, and to send data it takes about 45 minutes. The system should tell Thomas that within 30 minutes there are no passengers in gate 15 and the network quality is good in that gate. Hence, he can send this data in 12 minutes and in 10 minutes he can go and return to his door. So, he can ensure his activity (see Table 2). The RDF/XML description of the second situation of smart airport is presented in the box 4. It is conform to the proposed "IoTcontextOnt" ontology (see Figure 3).

In the box 4, we presented an RDF description of the smart airport situation. In this description, we used an "IoT-ContextOnt" as resource with the identifier "1245678" and we used the concepts "Location", "Person", "Time", "Activities", and "Preferences" as properties. In each property we defined resources as "Location" which regroups the properties "address", "terminal", "gate" and "passenger" the value assigned to the resource "address", "France", the value assigned to the resource "terminal", 3, the value assigned to the resource "gate", 3 and the value assigned to the resource "passenger" 12000. The person presented in this RDF description is "Passenger" named (property: "Name") "Thomas". This latter is "Audio-Video Engineer" (property: "Role"), He is located (property: "location")
in the "Gate 23" The activities of this person are described by the "Activities" property that regroups the resources: "When:210000", "Who: "Audio-Video Engineer", "Do: Send Files", "time Constrainte: Before 25 minute" and "Existed state: 45 minute". Finally the preferences of each person are described by "Preferences" property that re-
groups the person’s preferences. It is described by "Location: Gate 15" "Send Time: 15 minute", and "Round trip
time:10 minute".

Box 4.
In the second situation of smart home, we define the inputs and the outputs of the rules. The location of this activity is the airport and there is only one person named Thomas the role of this person is Audio-Video Engineer. His activity is to send large files in a limited time. The output of this rule is to define the location of the most suitable network to ensure this activity.

The box 5 presents the rule used to adapt the system in the second situation. It regroups the prefix used to define the namespase of the concepts; we present "prefixiotc" of the proposed ontology. In the input of the rule, we present the concepts "iotc:Name=Thomas" and "iotc:Location=Gate 23" as properties of the "perSys:Person" resource. Also, we use the concepts "iotc:who 'Audio-Video Engineer'" "iotc:when '210000'" "iotc:do 'Send Files'" "iotc:time_Constrainte 'Before 25 minute'" and "iotc:existed state '45 minute'" that exist in the "iotc:activity" property. The outputs of this rule is created by the update of the values assigned to the "iotc:Preference" property. It regroups the concepts; " rdf:location 'Gate 15'" " rdf:send_time '15 minute'" and "iotc:Round_trip_time '10 minute'". The result of the rule execution engine is to change values in the RDFDescription.

Box 5.
A smart ontology-based IoT framework for remote patient monitoring. The proposed IoTcontextOnt ontology is simple and easy to implement. This ontology is used to design an IoT system as a smart home or a smart city to ensure conceptual adaptation. It regroups a generic concept that can be used to describe any IoT system. The OpenIoT ontology describes abstraction of sensors and their integration with cloud computing concepts. OpenIoT ontology also uses SSN as a base to build upon required concepts for IoT applications and testbeds. Observation, Sensor, and Location. It further extends to define utility metrics for service level. IoTcontextOnt describes sensor in a smart situation, it concerns the modeling of sensors and the interaction of this latter with other objects in IoT system. IoT-O gives a conceptualization of the IoT domain, independent of the application, providing classes and relationships to link the underlying modules. It regroups the concept of four modules: Sensing module, Acting module, Lifecycle module, Service module, and Energy module. IoT-O ontology is designed to represent only connected devices networks without describing the user’s needs, network constraints, and location definition. In SAREF ontology, key concepts such as Actuator or Action are present but their representation is limited. Also, SAREF redefines the concepts present in multiple ontologies, and proposes alignments in an external textual document. The IoTcontextOnt regroups classes that can be instantiated in multiple situations; it presents the concepts which cover all descriptions of a situation within an IoT system. IoT-Lite ontology is a lightweight ontology to represent Internet of Things resources, entities, and services. IoT-Lite uses the most concepts defined in IoT system, but it did not take care of person’s characteristics and location. While most of the semantic models tend to describe the concepts in great details and represent various links in IoT systems, IoTcontextOnt represent only the most used concepts for data analytics in IoT applications. We have designed IoTcontextOnt with a clear purpose of defining only the most used terms when searching for IoT concepts in the context of data analytics. We studied the most common uses of IoT ontologies based on our experience with other IoT ontologies used by applications for data analytics.

5. Conclusion

In this paper, we have presented the IoT system (definition, architecture, applications domain, and challenges). Furthermore, we have presented a synthesis of the state-of-the-art works aiming to introduce ontologies into the Internet of Things. Based on our observations, we have proposed an IoT ontology named "IoTcontextOnt" that regroups essential information about the IoT system. Our ontology facilitates the merging of data collected in different domains for horizontal applications and allows the ontology to be an extendable core-domain. In addition, we proposed a conceptual adaptation architecture, which makes it possible to prepare the data to develop an IoT system. Finally, we illustrated the interest of our ontology using a smart home and smart airport as a case study. Moreover, we used the concept of OWL-S ontology in the semantic design of IoT system situation. In future works, we aim on the one hand, to introduce semantic and generic rules for IoT systems and to apply our proposed ontology to other smart spaces and add other descriptors to ensure the security of our system on the other.

References


[34] Z.Y.C. W. D. Jinfeng T. Tianran, Using intelligent ontology technology to extract knowledge from successful project in IoT enterprise system, Enterprise Information Systems (2021), 1–27.


