JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 83-100 (2022)
DOI: 10.6688/JISE.202201_38(1).0005

Towards a New Semantic Model for Service-Based IoT Applications

SAMIR BERRANI[1,+], ALI YACHIR[1], SAÏD MAHMOUDI[2],
BADIS DJAMAA[1] AND MOHAMED AISSANI[1]
[1]*École Militaire Polytechnique*
Algiers, 16111 Algeria
[2]*Computer Science Unit, Faculty of Engineering*
University of Mons
Mons, 7000 Belgium
[+]*E-mail: samir.berrani@yahoo.fr*

The next weave of Mobile Ad hoc Networks and Wireless Sensor Networks stands for the Internet of Things which aims to build a new ecosystem integrating the physical world with the digital one. Such a concept combines smart objects and the Internet infrastructure to ensure the ubiquitous accessibility, monitoring, and management of IoT resources like sensors, actuators, appliances, places, people, *etc.* However, this new ecosystem is highly dynamic, heterogeneous with an endless number of incorporated entities, with the same or different properties, evolving in a context that changes continuously. These challenging features may considerably compromise the IoT solutions' data interpretation, processing capacity, scalability, and the reuse and redundancy management of their entities. That is why we propose a lightweight, generic, and modular semantic model for designing IoT applications based on service composition. This model rests on Semantic Web Technologies and Service-Oriented Approaches. It provides a practical semantic description of IoT resources, such as sensors, actuators, services, environments, user requests, *etc.* It also meets the context-awareness and scalability properties and enhances the reuse and redundancy management of IoT entities. The proposed use-case scenario illustrates the interest, feasibility, and suitability of this model. Further conducted tests also show its performances and its high searching and querying ability.

Keywords: wireless sensor networks, internet of things, semantic model, service description, service discovery

1. INTRODUCTION

The second decade of the 21st century has witnessed the emergence of the Internet of Things (IoT). This concept aims to improve human beings' daily lives by transforming their surroundings into smart environments using billions of hybrid connected devices, such as sensors, actuators, *etc.* [1]. Such connected devices stand for Mobile Ad hoc Networks (MANET), Wireless Sensors Networks (WSN), Sensor/Actuator Networks (SANET), and independent rich nodes. By connecting such sub-networks and nodes to the Internet, contextual monitoring applications make homes, transportation, cities, *etc.*, smarter and dynamically adapting to their context [1].

However, the industrial and academic players involved in developing IoT solutions remain facing a big heterogeneity spread over three levels, perception, networking, and application-level [1, 2]. The perception heterogeneity is mainly due to the high number of

Received October 1, 2020; revised December 29, 2020 & February 18, 2021; accepted March 23, 2021.
Communicated by Changqiao Xu.

IoT solutions and their supporting technologies' producers. It is also due to the lack of standards and non-compliance with the existing ones. The perception heterogeneity is typically overcome through the Web of Things (WoT). Such a vision aims to abstract the IoT devices' functionalities as services. This abstraction supports the connection of IoT devices to the Web as well as the use of the common application layer protocols, such as HTTP and CoAP [3]. The network heterogeneity results from the various types of networks used by IoT solutions like Bluetooth, Zigbee, Wifi, LoRA, SIGFOX, *etc.* To surmount this issue, several approaches are proposed based on the gateways' concept [4]. These proposals ensure the federation of various network protocols. They also support bi-directional communication between different protocol stacks, for example, between IP and non-IP protocols.

Besides these issues, IoT solutions are hurting another issue that accounts for the service description's heterogeneity. The development of innovative cross-domain IoT applications based on service composition is therefore strongly compromised. Semantic web technologies have been primarily used to tackle this concern. They support building ontologies that provide semantic to service description and thus cope with this hinder. However, existing IoT semantic models ignore some primordial concepts and relationships which affect directly or indirectly the overall effectiveness of solutions that use them. Thus, designing such semantic models remains a crucial issue [5-8], particularly: How to enhance the ontologies' processing? How to meet the context-aware description of IoT resources, especially IoT services? How to fulfill the increasing scale nature of IoT solutions? How to improve the IoT resources' reuse and redundancy management?

This work presents a generic and modular semantic model for designing IoT applications based on service composition. It extends our previous knowledge model [9], which stands for a kickoff ontology. By completing the development process of this primary ontology, we have built the current model. Such a version supports the semantic description of applications, interest domain, entities like spaces and objects, devices, services, user requests, *etc.* It also meets the context-awareness and scalability properties and enhances the IoT concepts' reuse and redundancy management. The hierarchy of the model's concepts and their relationships enable IoT applications to improve semantic searching and querying ability. They further allow them to perform dynamic service discovery, automatic service selection, and flexible service composition.

The rest of this manuscript is organized as follows: Section 2 provides a brief review of the most recent and relevant IoT/WoT models. Section 3 introduces the proposed semantic IoT model. Section 4 presents an evaluation of the model's semantic searching and querying ability (the resource discovery). Finally, a conclusion is generated, and some future works are suggested in Section 5.

2. RELATED WORK

During this decade, several semantic models have been developed for the IoT domains. One of the most significant and popular knowledge models for describing IoT systems is the Semantic Sensor Network (SSN) ontology [10]. This model provides the semantic representation of sensors with their properties, stimuli, observations, *etc.* However, SSN neglects some IoT entities, such as actuators, processing boards, identifiers, and services. SSN is also seen as a heavyweight ontology for IoT systems. Various works are built upon this model to overcome its limits. The W3C group has defined a flexible and lightweight vocabulary for Sensors, Observations, Samples, and Actuators (SOSA) [11]. Based on the SSN core, SOSA represents entities, relations, and activities involved in sensing, sampling, and actuation. The IoT-A reference model also instantiates and extends the SSN device and sensor concepts to define core concepts for the IoT, namely Resources, Entities, and Services [12]. IoT-Lite is another generic lightweight ontology for the IoT resources' description [13]. This model improves the processing time of the resources' description and discovery. It further enhances the requests' resolution process. E. Mansour *et al.* [14] have also extended the representation of sensors, sensory data, and deployment environments of the SSN ontology. Such a model aims to address the high diversity of sensors, sensory data, platforms, and applications.

The above works require specific concepts for stream annotation and aggregation. This issue is addressed in [15], where T. Elsaleh *et al.* have extended SOSA ontology. They have annotated IoT data streams with simple temporal and value properties. A. Kamilaris *et al.* [16] have designed and developed a search engine based on web crawling for the Semantic Web of things. This solution includes a scalable and flexible process supporting the web device discovery, including their exposed services. R. Hafizur *et al.* [17] have proposed a lightweight ontology (LiO-IoT) for the IoT solutions using machine learning techniques. It encompasses only the most useful IoT concepts defined in SSN and IoT-Lite. This model provides semantic interoperability among heterogeneous devices and applications. It significantly reduces query response time compared to the existing heavy and complex ontologies. A. Yachir *et al.* [18] have presented a semantic approach based on ontological techniques combined with description logic. This model supports the semantic description of IoT devices, IoT services, and user requests.

N. Seydoux *et al.* [19] have proposed a core-domain modular IoT-Ontology (IoT-O) to describe the connected devices and their relationships with their environments. In such research, an open and dynamic knowledge representation for evolving IoT systems is provided. This ontology meets the self-adaptation property over time, depending on the changing state of the world. Apart from that, J. Swetina *et al.* [20] have set out a generic ontology, called oneM2M, which supports the discovery and remote control of devices and their services. oneM2M stands for an open standard service platform for M2M communications and IoT systems. Based on this standard, M. Ben Alaya *et al.* [21] have extended it to support semantic data interoperability. By doing so, the IoT-O's core concepts are used to describe M2M devices, including their exposed services. Similarly, the European Commission and the European Telecommunications Standards Institute [22] have developed a standard for smart appliances, Smart Appliances REFerence – SAREF. This modular and domain-independent semantic layer model is built upon the most relevant existing IoT ontologies. SAREF presents a seamless description, discovery, and remote control for smart appliances through the IoT service networks.

The W3C's WoT working group [23] has proposed a thing-centric formal model and a standard representation for WoT. The WoT Thing Description provides a semantic description of WoT-resources, including their metadata and interfaces. Based on this model, V. Charpenay *et al.* [24] have elaborated a new vocabulary. It encompasses the core WoT resources, thing Description, and Interaction. M. Noura *et al.* [25] have developed a WoT Description Language ontology. It supports End-User Development for WoT using a goal-oriented technique. This formalism provides the key concepts for describing WoT-resources and AI-planning for automatic WoT-composition. Previously in [9], we have proposed a primary modular semantic model for service-based applications in IoT environments. This initial prototype puts forward the general architecture of the expected IoT ontology with some illustrations. However, it remains to refine this first version to enhance its quality and build a comprehensive and efficient model. It also remains to evaluate and compare its performances with similar models.

IoT Middlewares and applications use ontologies to provide interoperable services. The performance of these processes strongly depends on the efficiency and completeness of their ontologies. A comprehensive and practical ontology improves its querying ability. Up-layer software thus improves its implemented business processes too. To complete this review, a comparative study is performed. It is based on three criteria that point out the features and limits of the reviewed models. They, therefore, depict their interest in IoT solutions. The first criterion stands for the model's expressiveness. It portrays the models' ability to describe the essential IoT systems' actors and entities, including their primary relationships. The second criterion represents the context-awareness property. A context stands for any information that enriches sensory data to be interpretable by different IoT applications [26]. It describes a situation, a place, a person, or anything that characterizes an environment. The location of targets and devices is vital information for context-aware computing. The interactions between devices strongly depend on their location and surroundings. Different methods are defined for performing context-aware systems [26]. These techniques mainly rest on some system's requirements and expectations that should be considered: (1) The targets and the used devices to observe and control them; (2) The targets' exact location and their relative locations regarding their neighboring targets; (3) The devices exact location and their relative location regarding their attached targets; (4) The devices' shared functionalities; (5) The time aspect. The third criterion accounts for IoT concepts' reuse and redundancy management. The reuse of ontology concepts improves the sharing and exchange of knowledge. The redundancy management enables the system's self-healing by offering the possibility to replace failing components with their equivalents.

Table 1 encloses a comparative study of the reviewed semantic models for IoT systems based on the above criteria. As to the model's expressiveness property, most of the models miss some primordial IoT concepts, except those proposed in [9, 11, 12, 21], which are quite comprehensive. Regarding the context-awareness property, the majority of models consider the domain business, time aspect, and the exact location of targets. They also support the tracking of devices based on the location of their attached targets. The models introduced in [9, 16, 17] further meet the location of services based on devices' location. Concerning the third comparison criterion, all reviewed models consider the concepts' reusing of target, device, and ontology. However, the redundancy management property is not mentioned whether they consider it or not. In this work, we aim to complete the model's development process proposed in [9] to cope with the above shortcomings. The expected model should meet the following expectations: Consider the composite concept of applications, devices, and services; Refine existing relationships between concepts and define new ones: *hasUser: (application, user), performs: (application and request), formulates: (user, request), hasProperties: (device and interest domain), holds: (target and device), hasSubject: (request and target), isLinkedTo: (request and service), isSatisfiedBy: (request, composite service), hasExpectedData: (request, domain ontology), hasSubject: (request,* *target), etc.*; Refine the description of targets, devices, and services, especially refining the description of their properties using the domain ontology of the application; Take into account the redundancy management of IoT concepts, ex. composite services that meet user requests. The final model should fit a wide range of IoT/WoT systems, especially applications based on service composition.

Table 1. Comparative study of the reviewed models.

Nº	Model	Model's expressiveness	Context-awareness	RM&R
1	[10]	L (T,Sr,DO)	M (DO,EL(T,D),Time)	L
2	[11]	L (T,Sr,Act,DO)	M (DO,EL(T,D),Time)	L
3	[12]	H (A,T,Sr,Act,Id,AS,DO,D+)	L (DO,Time)	M
4	[13]	H (A,T,Sr,Act,Id,AS,DO,D+)	M (DO, EL (T),Time)	L
5	[14]	L (T,Sr,Act,DO)	M (DO, EL (T,D),Time)	M
6	[15]	M (A,T,Sr,AS,DO,D+)	M (DO, EL (T),Time)	M
7	[16]	L (T,Sr,DO,D+)	L (DO, EL (T),Time)	L
8	[17]	M (T,Sr,Act,Id,AS,DO)	H (DO, EL (T,D,AS),Time)	L
9	[18]	M (T,Sr,Act,Id,AS,DO,D+)	H (DO, EL (T,D,AS),Time)	M
10	[19]	M (T,Sr,Act,Id,AS,DO)	M (DO, EL (T,D),Time)	L
11	[20]	M (T,Sr,Act,Id,AS,DO)	M (DO, EL (T,D),Time)	L
12	[21]	M (T,Sr,Act,Id,AS,DO)	M (DO, EL (T,D),Time)	M
13	[22]	H (T,Sr,Act,Id,AS,DO,T+,D+)	M (DO, EL (T,D),Time)	M
14	[23]	M (T,Sr,Act,AS,DO,T+,D+)	M (DO, EL (T,D),Time)	M
15	[24]	M(T,Sr,Act,AS,DO,T+,D+)	M (DO, EL (T,D),Time)	M
16	[25]	M(T,Sr,Act,AS,DO,T+,D+)	M (DO, EL (T,D),Time)	M
17	[9]	H– (A,T,Sr,Act,Id,PB,AS,R,DO,D+,T+)	H- (DO, EL (T,D,AS),Time)	H–

Model's expressiveness: Domain ontology DO, Application A, Target T, Target's physical properties T+, Device (Sensors Sr, Actuators Act, Processing Board PB, and Identifier Id), Device's functional properties D+, Atomic Service AS, User U, Request R; **Context-awareness:** DO, Time aspect, Exact and Relative location (EL & RL) of targets, devices, and services; **RM&R:** Redundancy management and reuse of IoT concepts; **Appreciation:** H: High, M: Medium, L: Low, N: Not considered, H−: High but the model is in its first round of the design process; it needs more refinement, improvement, and evaluation.

3. DESCRIPTION OF THE PROPOSED SEMANTIC IOT MODEL

IoT systems' nature is hybrid and heterogeneous, featured by a strong interaction between their software and hardware parts. We have used the Systems Modeling Language (SysML) and the Use-Case Modeling Approach (UCMA) to design a comprehensive systemic model for IoT systems. This model is used as an elementary study to develop an ontology for IoT service-based applications using the On-To-Knowledge methodology. The expected semantic model aims to improve the IoT services' description, discovery, selection, and composition process. It consists mainly of an application, a target, a device, a service, a request, and an interest domain. The defined relationships between the previous entities improve their reuse. Therefore, they enhance productivity, effectiveness, and the development's cost and time.

This model is also featured by high flexibility, allowing designers to choose the appropriate public/private sub-ontologies. An IoT application definition implicitly requires the specifications of its actors, entities, and their relationships. Fig. 1 shows that the proposed semantic model comprises six sub-descriptions: applications, interest domains, targets, devices, services, and requests. The IoT service-based application, bloc-A, defines IoT applications where the interest domain description, bloc-B, is used to specify their domain ontologies. Likewise, an IoT application manages IoT targets defined by the IoT target description, bloc-C. This module allows designers to define targets and their possible relationships. IoT devices, such as sensors, actuators, and processing boards, are defined using the IoT devices description, bloc-D. The IoT service description, bloc-E, describes services provided by IoT devices. The definition of the interactions between users and IoT applications is performed via the user request description, bloc-F. This module essentially defines requests through three parameters: an IoT application, subjects (targets), and an expected result or action (output). The semantic properties of targets, devices, services, and requests are specified using their domain ontology. These specifications define the targets' physical properties, the devices' functional properties, the services' inputs/outputs, and the requests' expected outputs.

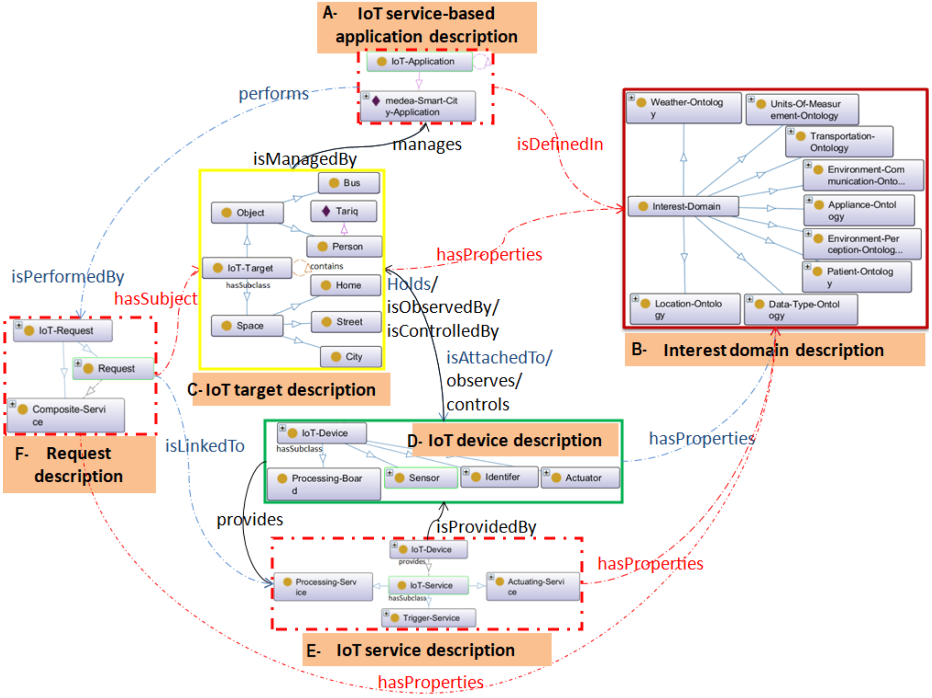


Fig. 1. General overview of the proposed semantic model (**NB**: Concepts and relationships with blue color are new; those with red color are updated compared to the initial model [9]).

3.1 Use Case Scenario: A Restricted Smart City

A restricted smart city scenario is proposed to illustrate the semantic model. It is supposed that Maria is a teacher who lives in a small city called Medea. She owns an e-wallet, which allows her to perform the electronic payments of several services, such as mobility, culture, leisure, and health service. Tariq is another actor in the proposed scenario. He is an old and partially sighted man who owns both an e-wallet too and an intelligent cane. This latter helps Tariq to move autonomously inside Medea. It provides him with his environment's ambient temperature and its Global Positioning System (GPS) location. Transportation inside Medea is mainly performed by buses. They are equipped with several smart devices like smart gates, air-conditioners, GPS, and ambient temperature sensors (inside/outside). The introduced targets like Tariq, Maria, and Buses stand for mobile WSAN. These networks are continually moving, interfering, turning off and on, *etc.* Thus, new larger and smaller networks are composed and broken down steadily.

3.2 IoT Application Description

IoT application description is constituted of an interest domain, targets, and requests, as shown in Fig. 2. Three relationships are defined between an IoT application and the above entities, namely, R1: isdefinedIn (IoT application, interest domain), R2: manages (IoT application, target), R3: perfroms (IoT application, request). R1 defines the domain ontology of an IoT application. R2 defines the targets managed by an IoT application. R3 defines the user requests performed by an IoT application. Besides, an interesting relationship, R4: "isComposedOf", is defined between IoT applications. Thus, the composition of applications is supported, and therefore their reuse is improved. These features enhance the application development's efficiency, productivity, insurance, cost, and time. For example, a restricted smart city application can be defined via the composition of other applications of the same city, like smart-transportation, building, and hospital.

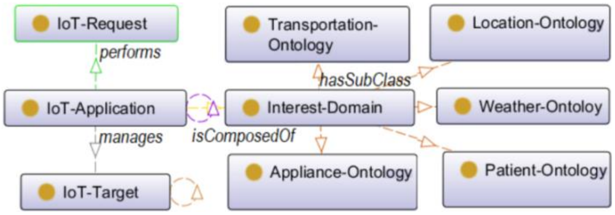


Fig. 2. Description of IoT application and interest domain.

3.3 Interest Domain Description

Interest domain ontology improves interoperability and interaction among the different entities of an IoT application. Fig. 2 shows the architecture of the proposed interest domain of the use-case scenario introduced in 3.1. The chosen domain ontologies should support describing the targets' physical properties, the devices' functional properties, the services' inputs/outputs, the requests' expected results, *etc.* Designers should choose and map the relevant domain ontologies to define an IoT application domain based on the above requirements. These domain ontologies can be public or private. Once the interest-domain concept maps the selected domain ontologies, they will be added to the application structure. This allows the designers of IoT applications to reuse the selected domain ontologies. In use-case 3.1, we have defined the "Medea-Smart-City-Application" (MSCA). This application manages several targets, such as Tariq, Bus-01, Home-01, *etc.* Tariq is a person who has several physical properties, such as a body-temperature, an address, and a

GPS location. Bus-01 is a vehicle described by speed, GPS location, capacity, and passengers. Home-01 is a home defined through an ambient temperature, an address, *etc*. MSCA's interest domain comprises five domain ontologies: a Transportation-Ontology, a Location-Ontology, a Weather-Ontology, a Patient-Ontology, and an Appliance-Ontology. These domain ontologies support the properties' definition of the MSCA's entities, including targets, devices, services, *etc*.

3.4 IoT Target Description

An IoT target stands for a smart space or object. This generic entity can be refined by more accurate concepts belonging to internal or external sub-ontology. Fig. 3 shows the description schema of the IoT targets related to use-case scenario 3.1. IoT targets are further featured by a set of physical properties defined using the interest domain ontology's semantic concepts. For instance, the semantic concepts "ambient-temperature" and "GPS-location" represent the physical properties of "Medea city", "Street-01", "Home-01", and "Bus-01". This specification enables several analyses, such as the classification and selection of IoT targets based on their physical properties.

Besides, an IoT target can contain sub-targets. In smart transportation, a bus transports passengers. The bus and passengers are targets. In this case, the bus contains passengers. The transitive relationships "contains" and "isContainedIn" hierarchize targets. This structure supports the targets' tractability. It also supports the deduction of meaningful information that can be useful for making decisions.

Furthermore, a new concept is introduced, the target branch. Such a concept is defined as the union of a given target and its sub and sup targets. It is mainly used to perform a context-aware resource discovery. As shown in Fig. 3, the target branch of "Bus-01" comprises, its sub-targets, "Tariq" and, its sup-targets, "Street-01" and "Medea-city".

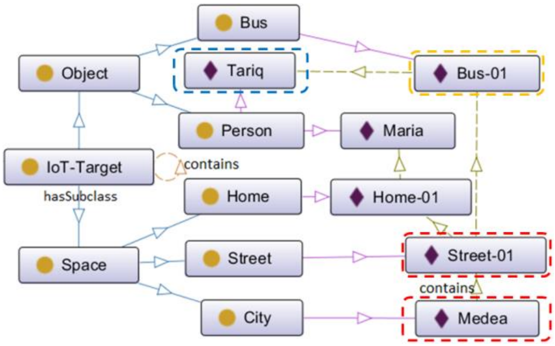


Fig. 3. Description of IoT targets and illustration of the branch's concept.

3.5 IoT Device Description

The IoT devices are nonstandard computing entities featured by their constrained resources. Four types of IoT devices are distinguished according to their functionalities, sensors, actuators, processing-boards, and identifiers. Fig. 4 shows the above subclasses. The composition process of IoT devices is supported. This process allows designers to compose objects according to their supplied functionalities to meet the client's requirements and needs. For example, a Raspberry processing board (Pi) or a NodeMCU Development Board (ESP-12E) can incorporate an ambient temperature sensor, an obstacle detector, a light signal, *etc.* IoT devices are generally described by an identifier (IP, URI, serial number, *etc.*), a name, a brand, a model, a producer, a description, *etc.* They are also featured by two functional states: online or offline. It seems useful for an offline object to set its provided functionalities (services) offline automatically.

Such a way guarantees the consistency of the system (object and its provided functionalities). IoT devices like sensors, processing boards, and actuators perform specific tasks described through semantic concepts of the application's interest domain ontology. For example, the intelligent cane introduced in 3.1, "Intelligent-cane-01", provides its GPS location, the ambient temperature of its environment, a light signal (visual alarm), an obstacle meter, and an obstacle warning. To represent these functionalities, four relationships are created between "Intelligent-cane-01" and the following semantic concepts: "GPS-location", "ambient-temperature", "obstacle", and "visual-alarm".

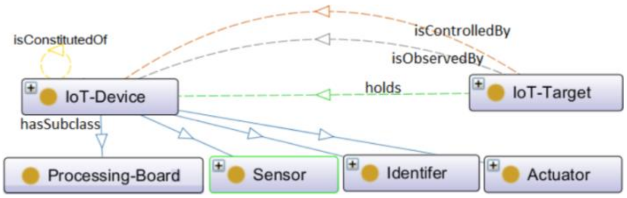


Fig. 4. IoT devices' sub-classes.

3.6 IoT Service Description

An IoT service implements a physical or a logical function that provides contextual data or applies actions on its attached targets. The contextual data stands for a specific phenomenon's observations/measurements, such as the home's ambient temperature. However, the application of a particular action enables the control of its related phenomenon. For example, keep the temperature in an office at 25 Celsius. IoT service description provides a handy and lightweight definition for services. It encompasses the IoT service direct and indirect relationships (ObjectProperties) with the other entities like devices, requests, and the concepts of the interest domain ontologies. It also includes the definition of the services' features (DataProperties). The proposed description simplifies the service selection task and reduces the complexity of the service composition process.

The most relevant parameters proposed to define an IoT service are as follows: (1) Identifier: Resources, including services on the Internet, are identified by URIs; (2) Category: Based on the IoT device categories, three types of service are distinguished: trigger, processing, and actuating; (3) Inputs/Outputs: Besides the URI of a given service, its parameters are required to invoke it. A parameter is defined through a semantic concept, a syntactic type, a unit, an abstract value, a data format, and a timestamp.

3.7 IoT Request Description

The IoT request description enables users to define their requests, which are performed by IoT applications. Relationship between requests and their owners are defined, "isFormulateBy/formulates". Therefore, the model can collect the user's activities and deduce their preferences. Requests are equivalent if and only if they are defined through similar output data and subjects. As shown in Fig. 5, the IoT request schema is mainly composed of a request, a composite service, and a service sequence entity. The request entity is described through subjects (IoT targets) and output data (expected data/action). This latter is defined using the concept of parameter and the semantic concepts of the interest domain ontology, as seen in the above sub-section.

A request resolution process performs composite services that satisfy the users' requests. Such solutions are represented using the Composite-Service entity. They are defined via a URI, nature, a state, an invocation mode, input/output parameters, pre/post-conditions, *etc.* A composite service entity is defined through a sequence of services. Each node of this structure should be linked to a unique service. Thus, the composite service can be reproduced without any additional processing. Therefore, it will be ready for reusing immediately. This enhances the reusability of requests and their related composite services. Besides, when a composite service fails during its invocation, one of its equivalent composites will be selected and called automatically. Then, a self-healing procedure can be used to fix the failed composite. This mechanism finds the failed components (services). Then, it replaces them with their redundant. Finally, it updates the composite state, including its services. These interesting functionalities are supported primarily because our model defines relationships (isSatisfiedBy/satisfies) between user requests and composite services that meet them.

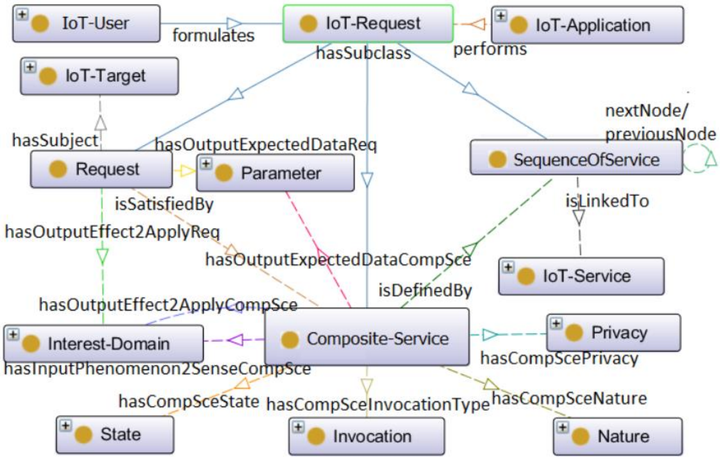
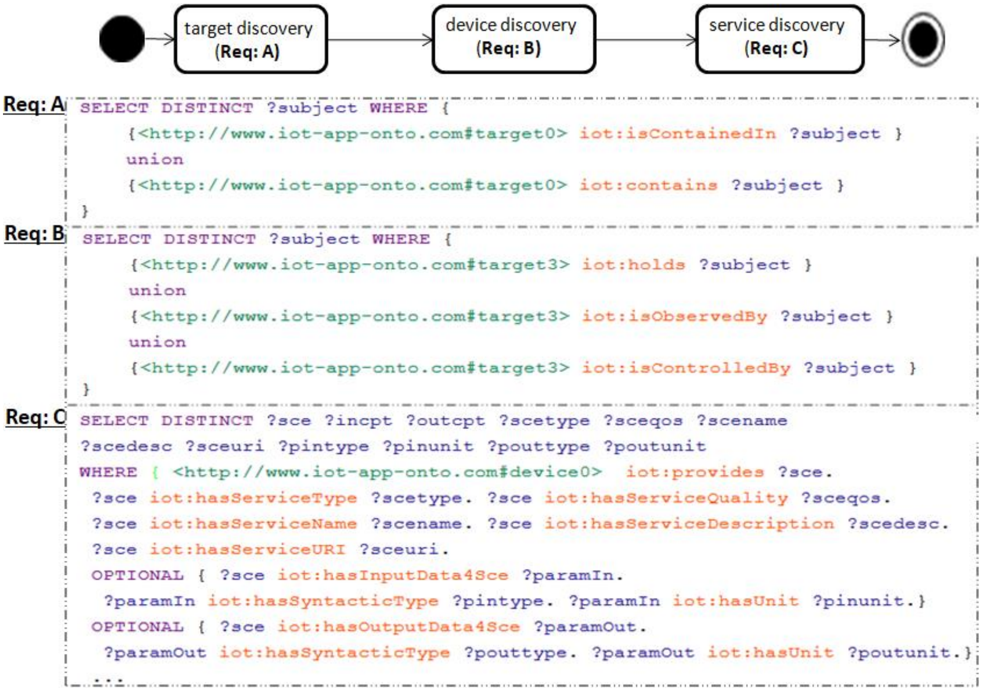


Fig. 5. Description schema of an IoT request.

4. IOT RESOURCE DISCOVERY EVALUATION

This section aims to show the high searching and querying ability of the proposed model. To this end, we have chosen the IoT resource discovery process. Such a process searches and selects the most appropriate resources to meet a given user request. The activity diagram shown in Fig. 6 depicts that three sub-processes make up the IoT resource discovery process. The first sub-process stands for the target discovery, which performs the user request’s subject branch. Request A of Fig. 6 illustrates the SPARQL query that carries out the target branch of a given target, <...*#target0*>. The second sub-process uses the discovered targets to list their attached devices, representing the discovered devices’ set. Request B of Fig. 6 exemplifies the SPARQL query that performs the attached devices to a given target, <...*#target3*>. The last sub-process stands for the service discovery, which lists the provided services of the discovered devices. Request C of Fig. 6 points out the SPARQL query that enumerates the provided services of a given device, <...*#device0*>.



```
SELECT DISTINCT ?subject WHERE {
    {<http://www.iot-app-onto.com#target0> iot:isContainedIn ?subject }
    union
    {<http://www.iot-app-onto.com#target0> iot:contains ?subject }
}
```

```
SELECT DISTINCT ?subject WHERE {
    {<http://www.iot-app-onto.com#target3> iot:holds ?subject }
    union
    {<http://www.iot-app-onto.com#target3> iot:isObservedBy ?subject }
    union
    {<http://www.iot-app-onto.com#target3> iot:isControlledBy ?subject }
}
```

```
SELECT DISTINCT ?sce ?incpt ?outcpt ?scetype ?sceqos ?scename
?scedesc ?sceuri ?pintype ?pinunit ?pouttype ?poutunit
WHERE { <http://www.iot-app-onto.com#device0>  iot:provides ?sce.
 ?sce iot:hasServiceType ?scetype. ?sce iot:hasServiceQuality ?sceqos.
 ?sce iot:hasServiceName ?scename. ?sce iot:hasServiceDescription ?scedesc.
 ?sce iot:hasServiceURI ?sceuri.
 OPTIONAL { ?sce iot:hasInputData4Sce ?paramIn.
  ?paramIn iot:hasSyntacticType ?pintype. ?paramIn iot:hasUnit ?pinunit.}
 OPTIONAL { ?sce iot:hasOutputData4Sce ?paramOut.
  ?paramOut iot:hasSyntacticType ?pouttype. ?paramOut iot:hasUnit ?poutunit.}
 ...
```

Fig. 6. Illustration of the discovery process based on SPARQL queries.

4.1 Abstraction of an IoT Environment

This section introduces the IoT environment concept. To understand the idea behind this, let us reconsider the use case scenario introduced in 3.1. MSCA manages independent and dependent targets. As shown in rectangle A of Fig. 7, van-01 and street-02 are independent. There is no relationship between the above targets. However, car-01 and street-02 are dependent since car-01 is on street-02. IoT devices attached to these targets observe and control them. Such components expose their functionalities as services. By doing so, cross-domain applications can use these services without restrictions.

Similarly, based on this example, we can constitute an IoT environment for testing the proposed model’s performance. Such an environment is represented by an ontology using the proposed model. This semantic schema defines a set of independent/dependent targets. It also specifies the attached devices of each target. Further, it names the provided services of the above devices. To these entities, the test ontology encompasses an interest domain ontology, units, parameters, *etc*. Rectangle B of Fig. 7 illustrates the abstraction of an IoT environment.

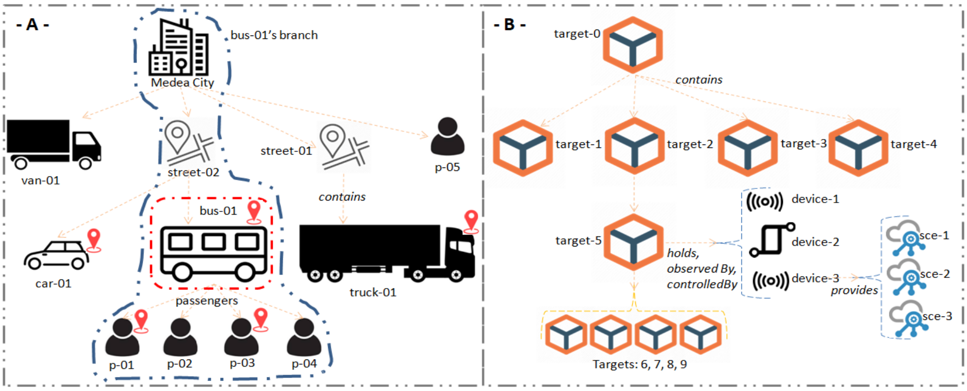
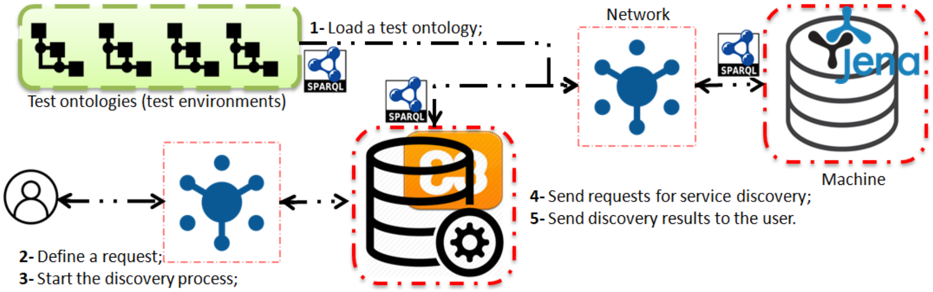


Fig. 7. Illustration of an IoT environment abstraction.

4.2 Experimental Methodology Protocol

This evaluation aims to study the IoT resource discovery time depending on the IoT environment dimension and the request subject branch length. The dimension of an IoT environment accounts for the number of its targets, devices, and services. The request subject branch length stands for the number of targets in the branch. Besides, we have developed a generator to generate RDF triple-stores, called test ontology, representing an IoT environment concordant with the proposed model. This generator provides the possibility to specify the ontology's dimension, the number of targets, devices, services, semantic concepts, syntactic types, units, *etc*. The generated requests can be defined with the composite services that fulfill them. The numbers of targets, devices, and services are equal for each test ontology. This choice is made to simplify the evaluation task. The IoT concept relationships defined by the proposed semantic model are considered. This assessment is performed using a laptop with windows 10 professional, Intel(R) Core(TM) i7-1500U CPU @ 2.70GHz, 8Go RAM, Oracle Java 1.8.0 221.

As shown in Fig. 8, we have followed an experimental protocol made up of 5 steps. First, we load the test ontology on the Apache-Jena server. Then, we define a request, in-



1- Load a test ontology;

2- Define a request;
3- Start the discovery process;

4- Send requests for service discovery;
5- Send discovery results to the user.

Fig. 8. Illustration of the evaluation protocol.

cluding its subject. Next, we start the discovery process. We send sequentially 1000 requests to discover the most relevant services that can meet the request. We measure for each evaluation, the target discovery time, the device discovery time, and the service discovery time. Finally, we calculate the average value and the standard deviation, the error bars, of the obtained measurements. By doing so, we obtain a more accurate and understandable evaluation.

4.3 Impact of the Environment Dimension on the Resource Discovery Time

This subsection studies how the ontology dimension affects the resource discovery time. The length of the test request subject's branch is fixed to ten. Ten test ontologies are generated whose dimensions are equal to 1500, 3000, 4500, 6000, 10000, 20000, 30000, 40000, 50000, and 60000 nodes. The test request is common for all evaluations. Fig. 9 depicts the test ontology dimension's impact on the needed time to discover targets, devices, and services. It is noticed that the discovery plots of targets and devices are linear and constant. However, the service discovery curve increases by increasing the environment's dimension. The standard deviations of the performed evaluations are due to the networks' load, the OS memory management, the OS CPU scheduling, the request management processes of both Apache Jena (Semantic Database System Management) and Apache HTTP Server (XAMPP - Application Server System Management), *etc.*

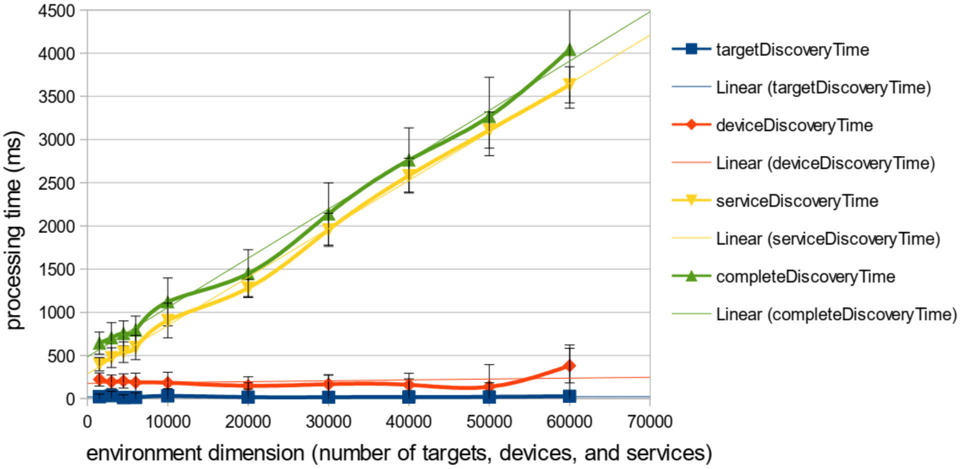


Fig. 9. Impact of the environment dimension on the resource discovery time.

4.4 Impact of the Request Subject Branch Length on the Resource Discovery Time

This subsection studies how the request subject branch length affects the resource discovery time. The dimension of the test ontology is fixed to 3000 nodes. The request subject branch length varies from 10 to 100 targets, 10, 25, 50, 75, and 100. Each target of the above branch contains one device, and in turn, each device provides one service. Fig. 10 shows the impact of the request subject branch length on the IoT resources' discovery time, including targets, devices, and services. It is observed that the target discovery time is neglected. It is further noticed that the device discovery time is less important than the service discovery time, and both increase with the increasing of the request subject branch length.

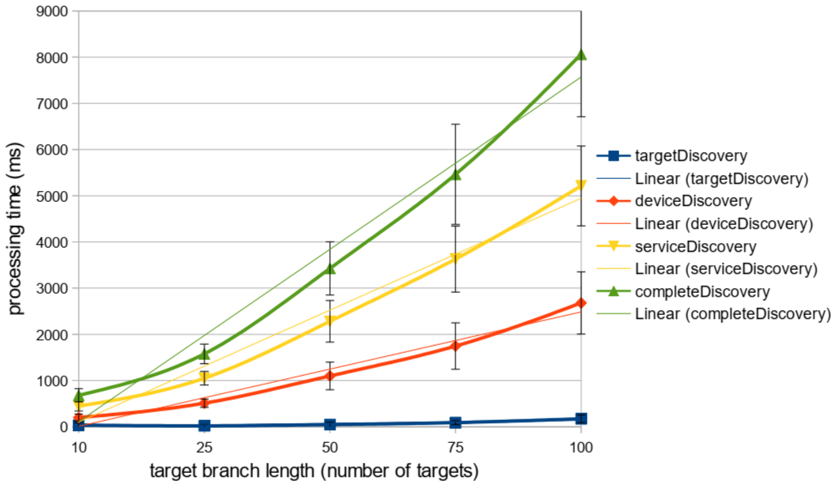


Fig. 10. Impact of the request subject branch length on the resource discovery time.

4.5 Comparison

Among the reviewed works, only LiO-IoT [17] provides an assessment that studies how the number of nodes affects the response time of a given request for three ontologies, SSN [10], IoT-Lite [13], and LiO-IoT [17]. It is noticed that the response time for a given request increases when the number of nodes increases for all models. The same remark is stated in Section 4.3, where it is studied how the number of nodes affects the IoT service discovery time. In such an assessment, the subject branch of the test request encompasses 10 targets. Each target contains one device, and that device provides one service. To perform the IoT service discovery process, 21 requests are executed (1 request to get target branch, 10 requests to get the attached devices of the branch's targets, and finally, 10 requests to get the provided services of the discovered devices.). As shown in Fig. 9, in test environments of 30000, 40000, 50000, and 60000 nodes, the service discovery requires 2139, 2763, 3268, and 4044 (ms), respectively. Thus, the average times required to execute a request are 102, 132, 156, and 193 (ms), respectively, according to the above results.

Based on the evaluation presented in [17], the measured query response time for SSN, IoT-Lite, and LiO-IoT are 986, 106, and 162 (ms), respectively. These experiments are performed using the following test ontologies, SSN: 200 sensors, IoT-Lite: 200 sensors, and LiO-IoT: 200 sensors, 200 actuators, and 200 RFID (600 nodes). The processing speed of the processor is 83000 (MIPS). These results are explained by [17] as follows: (1) The SSN ontology's query response time is the greatest because it includes several unnecessary concepts. (2) The IoT-Lite ontology requires roughly less query response time than LiO-IoT because of its very abstract nature.

Table 2 shows the impact of the number of nodes on the request response time for SSN, IoT-Lite, LiO-IoT, and the proposed model. This latter provides encouraging performances due to the generic nature of the model, its efficient hierarchy, its reduced and sufficient concepts, and its high searching and querying ability.

Table 2. Impact of the number of nodes on the request-response time per request.

N°	Model	Ontology's size	Query response time (ms)
1	SSN [10]	200	986
2	IoT-Lite [13]	200	106
3	LiO-IoT [17]	600	162
4	Proposed Model	30000, 40000, 50000, 60000	102, 132, 156, 193

5. CONCLUSION AND FUTURE WORK

This paper presents a new lightweight, generic, and modular semantic model for designing IoT applications based on service composition. The existing semantic models partially address the context-awareness, scalability, IoT concepts' reuse, and redundancy management properties. To cope with these limits, we have first designed a SysML model for IoT systems, particularly those based on service composition, to identify their most relevant actors, entities, and relationships. Based on this conceptualization, we have developed a lightweight, generic, and modular ontology using the On-To-Knowledge methodology. This model considers the most significant IoT entities, actors, and IoT properties, especially the context-awareness, scalability, and the IoT concepts' reuse and redundancy management. The use-case scenario illustrates the interest, feasibility, and high searching and querying ability of the proposal. The performed tests of the IoT resource discovery present admissible and encouraging performances. As future work, we plan to develop a semantic Middleware for IoT applications based on service composition using this model. Such a solution encompasses several exposed services like a handy and comprehensive resource description and an end-to-end loosely-coupled request resolution process, including subprocesses like service discovery, selection, and composition.

REFERENCES

1. R. Hassan, F. Qamar, *et al.*, "Internet of Things and its applications: A comprehensive survey," *Symmetry*, Vol. 12, 2020, pp. 1674.
2. A. Khanna and S. Kaur, "Internet of Things (IoT), applications and challenges: A comprehensive review," *Wireless Personal Communications*, Vol. 114, 2020, pp. 1687-1762.
3. S. S. Mathew, Y. Atif, and M. El-Barachi, "From the Internet of Things to the web of things – enabling by sensing as-a service," in *Proceedings of the 12th International Conference on Innovations in Information Technology*, 2016, pp. 1-6.
4. S. Sinche, *et al.*, "A survey of IoT management protocols and frameworks," *IEEE Communication Surveys and Tutorials*, Vol. 22, 2020, pp. 1168-1190.
5. M. Harlamova, *et al.*, "A survey on challenges of semantics application in the Internet of Things domain," *Applied Computer Systems*, Vol. 21, 2017, pp. 13-21.

6. R. Hafizur and H. Md. Iftekhar, “A comprehensive survey on semantic interoperability for Internet of Things: State-of-the-art and research challenges,” *Transactions on Emerging Telecommunications Technologies*, Vol. 31, 2020, p. e3902.
7. S. Sengupta, J. Garcia, *et al.*, “A literature survey on ontology of different computing platforms in smart environments,” *CoRR*, 2018, Vol. abs/1803.00087.
8. M. E. Adam, “Usages of semantic web services technologies in IoT ecosystems and its impact in services delivery: A survey,” *International Journal of Computer*, Vol. 36, 2020, pp. 53-72.
9. S. Berrani, A. Yachir, B. Djemaa, and M. Aissani, “A semantic model for service description in the internet of things,” in *Proceedings of International Conference on Smart Communications in Network Technologies*, 2018, pp. 49-54.
10. M. Compton, P. Barnaghi, L. Bermudez, *et al.*, “The SSN ontology of the w3c semantic sensor network incubator group,” *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 17, 2012, pp. 25-32.
11. K. Janowicz, A. Haller, *et al.*, “SOSA: A lightweight ontology for sensors, observations, samples, and actuators,” *Journal of Web Semantics*, Vol. 56, 2019, pp. 1-10.
12. M. Bauer, N. Bui, J. de Loof, *et al.*, “IoT reference model,” in *Enabling Things to Talk*, Springer, Berlin, Heidelberg, ch. 08, 2013, pp. 113-162.
13. M. Bermudez-Edo, T. Elsaleh, *et al.*, “IoT-Lite: A lightweight semantic model for the internet of things and its use with dynamic semantics,” *Personal and Ubiquitous Computing*, Vol. 21, 2017, pp. 475-487.
14. E. Mansour, R. Chbeir, and P. Arnould, “HSSN: An ontology for hybrid semantic sensor networks,” in *Proceedings of the 23rd International Conference on Database Applications and Engineering Symposium*, 2019, pp. 1-10.
15. T. Elsaleh, S. Enshaeifar, R. Rezvani, *et al.*, “IoT-stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services,” *Sensors*, Vol. 20, 2020, p. 953.
16. A. Kamilaris, S. Yumusak, and M. I. Ali, “Wots2e: A search engine for a semantic web of things,” in *Proceedings of IEEE 3rd World Forum on Internet of Things*, 2016, pp. 436-441.
17. H. Rahman and M. I. Hussain, “A light-weight dynamic ontology for internet of things using machine learning technique,” *ICT Express*, Vol. 7, 2020, pp. 355-360.
18. A. Yachir, B. Djamaa, A. Mecheti, *et al.*, “A comprehensive semantic model for smart object description and request resolution in the internet of things,” *Procedia Computer Science*, Vol. 83, 2016, pp. 147-154.
19. N. Seydoux, K. Drira, N. Hernandez, *et al.*, “IoT-O, a core-domain IoT ontology to represent connected devices networks,” *Knowledge Engineering and Knowledge Management*, 2016, pp. 561-576.
20. J. Swetina, G. Lu, *et al.*, “Toward a standardized common m2m service layer platform: Introduction to onem2m,” *IEEE Wireless Communications*, Vol. 21, 2014, pp. 20-26.
21. M. B. Alaya, S. Medjiah, T. Monteil, *et al.*, “Toward semantic interoperability in onem2m architecture,” *IEEE Communications Magazine*, Vol. 53, 2015, pp. 35-41.
22. ETSI (2017), “SmartM2M; smart appliances; reference ontology and oneM2M mapping,” European Telecommunications Standards Institute, Ref:RTS/SmartM2M 103264v2, accessed November 11, 2020.
23. W3C (2020), “Web of Things (WoT) thing description,” accessed November 11, 2020.

24. V. Charpenay, S. Käbisch, and H. Kosch, “Introducing thing descriptions and interactions: An ontology for the web of things,” in *Joint Proceedings of SR+SWIT@ISWC*, Vol. 1783, 2016, pp. 55-66.
25. M. Noura and M. Gaedke, “Wotdl: Web of things description language for automatic composition,” in *Proceedings of International Conference on Web Intelligence*, 2019, pp. 413-417.
26. H. Chen, T. Finin, and A. Joshi, “An ontology for context-aware pervasive computing environments,” *The Knowledge Engineering Review*, Vol. 18, 2003, pp. 197-207.

Samir Berrani is currently a Ph.D. candidate at the Ecole Militaire Polytechnique (EMP), Algeria. His work focuses on applications based on service composition in the IoT environment. He is also interested in Model-Based Systems Engineering. He has obtained a Master’s degree in safety and security of software from Franche-Comté University in 2012. Previously, he has obtained his computer engineering degree from EMP in 2004.

Ali Yachir received the M.Sc. degree in Computer Science from Bejaia University, Algeria, in 2008, and the Ph.D. from USTHB University, Algeria, and UPEC University, Creteil, France, in 2014. Now, He is an Associate Professor at EMP of Algiers. His current research interests include intelligent ambient systems, wireless sensor networks, Internet and Web of things, semantic web technologies, and service-oriented architectures.

Saïd Mahmoudi is an Associate-Professor at the Faculty of Engineering of the University of Mons, Belgium. He is graduated from the University of Oran, Algeria. He received his MS in Computer Science from the LIFL Laboratory, University of Lille, France, in 1999. He obtained his Ph.D. in Computer Science at the University of Lille1, France, in 2003. His research interests include artificial intelligence, the internet of things, image processing, computer-aided diagnosis, *etc.*

Badis Djamaa is a Teacher-Researcher at EMP. He received a Computer science State Engineering Degree from EMP in 2011 and a Ph.D. degree from Cranfield University, the UK, in 2015. His current research activities include wireless sensor and actuator networks, lightweight security systems, (industrial) internet and web of things, semantic technologies, service-oriented architectures, and information-centric networks.

Mohamed Aissani is a Teacher-Researcher at EMP. He received a M.Sc. and a Ph.D. degree in computer science from USTHB University, Algeria, in 2002 and 2014, respectively. His current research interests include models, communication protocols, routing algorithms, energy consumption issues of wireless sensor and actuator networks, the Internet, and the Web of things.