

# Unified IoT Ontology to Enable Interoperability and Federation of Testbeds

Rachit Agarwal<sup>1</sup>, David Gomez Fernandez<sup>2</sup>, Tarek Elsaleh<sup>3</sup>, Amelie Gyrard<sup>4</sup>,  
Jorge Lanza<sup>2</sup>, Luis Sanchez<sup>2</sup>, Nikolaos Georgantas<sup>1</sup>, and Valerie Issarny<sup>1</sup>

<sup>1</sup> Inria-Paris, France,

<sup>2</sup> University of Cantabria, Spain

<sup>3</sup> University of Surrey, UK

<sup>4</sup> Insight Center for Data Analytics, NUIG, Ireland

{rachit.agarwal, nikolaos.georgantas, valerie.issarny}@inria.fr, {dgomez,  
jlanza, lsanchez}@tlmat.unican.es, t.elsaleh@surrey.ac.uk,  
amelie.gyrard@insight-centre.org

**Abstract.** After a thorough analysis of existing Internet of Things (IoT) related ontologies, in this paper we propose a holistic and a light-weight ontology that aims to achieve semantic interoperability among various fragmented testbeds (that store data in their proprietary format) in IoT domain. Such ontology currently finds its implementation in the EU H2020 FIESTA-IoT project that aims to provide federation and interoperability to testbeds using semantics. The proposed ontology leverages from the Noy et al. methodology of building ontology. With the aim of including more ontologies in the future, if needed, our ontology currently reuses core concepts from various popular ontologies and taxonomies, such as *Semantic Sensor Network* (SSN), *M3-lite* (a lite version of M3 and also an outcome of this study), *WGS84*, *IoT-lite* and *DUL*. In addition to this, we also introduce a set of tools that aims to help external testbeds adapt their respective datasets to the common ontology defined throughout this paper.

**Keywords:** Semantic Web of Things; Internet of Things; Semantic Web Technologies; Ontology; Interoperability; Federation; Testbed

## 1 Introduction

One of the most highlighting features of the Internet of Things (IoT) domain is the heterogeneity of the information that comes from the underlying devices. Such heterogeneity and openness brings lack of standards that every platform should have followed. Many testbeds owning the devices or applications interacting with the sensors, store such observations and other related data in their own proprietary format (according to [1], a testbed is “*an environment that allows experimentation and testing for research and development products. A testbed provides a rigorous, transparent and replicable environment for experimentation and testing*”). Thus, making a testbeds independent and isolated from others

that cannot directly interact with each other. Differences in the data format results into interoperability issues between the testbeds and much work has to be done in order to ensure the interoperability (excerpted from [2], interoperability is “*the ability of two or more systems or components to exchange information and use the information that has been exchanged*”). One has to understand different data formats and create the mapping. As a result, a testbed must understand other testbed’s format and create the mapping both while sending information and receiving information.

One method to achieve such interoperability is via achieving semantic interoperability by providing semantic models to the observations produced by the devices. In this paper, we propose a fully-fledged ontology that spans across all the necessities that a testbeds deal with. We leverage for our previous knowledge from the Linked Open Vocabularies for Internet of Things (LOV4IoT)<sup>5</sup> project. In LOV4IoT, there is a reference to 39 relevant ontologies. These ontologies are either in the domain of sensor networks or much broader domain of IoT. Nevertheless, some of the relevant existing ontologies promise interoperability but: *(i)* do not address the problem to describe the observation in an interoperable manner to ease the tasks such as reasoning, *(ii)* many ontologies are domain specific and can not be applied across domains, *(iii)* have missing concepts and do not suffice the needs of the measurements provided by the sensors, and *(iv)* many ontologies do not follow best practices making it hard to correctly interpret concepts.

In order to address above mentioned limitations, via this paper, we provide a unified semantic model that follow best practices. The semantic model is supported by set of tools like reference annotator and validator which testbeds can use to induce interoperability. For our ontology, the best practices are mainly applied in the building phase and support. Further, we are motivated not by creating an overloaded ontology but by creating a light-weight ontology that addresses interoperability issues and various aspects of IoT device observations. Our ontology is focused on the description of the underlying testbeds’ resource descriptions and the observations gathered from their physical devices (e.g. sensors). The current ontology is developed under and applied to H2020’s FIESTA-IoT project<sup>6</sup>, that aims to support federation and interoperability among different orthogonal testbed by means of the usage of semantic-based techniques. We name our ontology as **FIESTA-IoT** ontology.

Finally, this paper is structured as follows: Section 2 presents a thorough state-of-the-art analysis and provides all the necessary background knowledge about the IoT-related ontologies. Upon the knowledge acquired with this study, Section 3 introduces the *FIESTA-IoT* ontology that is build from: *(i)* various models tackled in Section 2, and *(ii)* new elements exclusively adapted for the ontology, currently available from *M3-lite taxonomy*. The ontology is supported via reference annotation and validation tool along with best practices and guidelines. In Section 4 we identify some potential uses of the ontology. Finally, we

<sup>5</sup> <http://sensormeasurement.appspot.com/?p=ontologies>

<sup>6</sup> <http://fiesta-iot.eu/>

conclude via Section 5 and presents open issues that will be addressed in the future.

## 2 Related work

Many IoT related surveys are available that study related ontologies [3, 4], however, these works do not show recent advancements. There are many ontologies that have been made available since then that specifically deal with IoT, sensors and other related domains. In this section we describe some of the IoT related ontologies available and address why a particular ontology is relevant to us.

Out of all available ontologies in the Sensor and IoT domain, only SSN [5] follows the best practices according to LOV4IoT and is also recommended by Linked Open Vocabulary<sup>7</sup> (LOV). SSN defined as a part of World Wide Web Consortium still lacks concepts to describe phenomenon sensed, the unit of observed value, location, and time leading to interoperability issues between non standard domain specific ontologies and SSN. Another ontology, IoT-A ontology [6], provides core concepts such as Service, and is mainly Service oriented. It reuses only *ssn:condition* concept. Further, it is complex, lacks usage of standard ontologies, and has redundancy issues. IoT-lite<sup>8</sup>, one of the adaptations of IoT-A, is much powerful and simpler than IoT-A. It uses concepts from SSN and extends it by addressing shortcomings of SSN. This ontology along with SSN defines most of the concepts used in our ontology and thus the concepts within IoT-lite are reused in our ontology. Further, various updates to IoT-lite have been done within the context of FIESTA-IoT. Yet other ontologies such as Open-IoT ontology [7], VITAL<sup>9</sup> and Hachem et al. [8], extends SSN. Open-IoT ontology introduces concepts like utility metrics for physical and virtual sensors, and services. VITAL extends Open-IoT ontology while [8] provides concepts for defining mathematical models for phenomenons. Such concepts though are not available in our current version of the ontology, but can be added.

IoT-O [9] is yet another example of an ontology that reuses concepts from SSN and is designed to support heterogeneity in IoT. It reuses concepts like `ssn:Device` and `ssn:ObservationValue` from SSN. Further, it also uses concepts from other LOV recommended ontologies, such as Semantic Actuator Network (SAN)<sup>10</sup>, DOLCE Ultra Lite (DUL)<sup>11</sup>, Time<sup>12</sup>, and Quantities, Units, Dimensions and Data Types (QUDT)<sup>13</sup>. In addition, IoT-O is well documented and uses Pellet reasoner for inference. The ontology rules are shared following the idea of Semantic based Linked Open Rules (S-LOR). IoT-O follows best

<sup>7</sup> <http://lov.okfn.org/dataset/lov/>

<sup>8</sup> <http://iot.ee.surrey.ac.uk/fiware/ontologies/iot-lite>

<sup>9</sup> <http://vital-iot.eu>

<sup>10</sup> <https://www.irit.fr/recherches/MELODI/ontologies/SAN>

<sup>11</sup> <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>

<sup>12</sup> <https://www.w3.org/TR/owl-time/>

<sup>13</sup> <http://qudt.org/schema/qudt#>

practices, integrates with oneM2M standardizations [10] and provides benefit of modularization.

oneM2M on one hand provides standards, but on the other, it also provides a base ontology [10] with the aim to help non-oneM2M compliant data models derive oneM2M concepts to describe their data model. As oneM2M is soon to become a standard, we intend to standardize our ontology and currently working on integrating our ontology with oneM2M.

Nevertheless other than the concepts within an ontology, it is also important to describe taxonomy. M3 ontology [11] is one such effort where various IoT related concepts from various ontologies are integrated with a unified taxonomy that describes concepts like *Domain of Interest*, *Physical Phenomena* and *Units of Measurement*. As M3 is heavy, a light-weight version of M3 is also proposed by us in Section 3.3 and used within our ontology.

Further, there are various other IoT related ontologies that are available, such as [12, 13]. Nevertheless, due to space constraints, we refrain ourselves in explaining all the available IoT related ontologies.

The main contribution of this work is the construction of a *light-weight ontology* that deals with IoT domain and guarantee a complete semantic interoperability. To achieve this, we have taken a set of core concepts from various well-known ontologies and integrated them to support the semantic description of IoT services/resources, observations and Virtual Entities that might come from the various underlying testbeds.

### 3 FIESTA-IoT Ontology: A Unified Ontology

Many of the existing ontologies are not well connected and are domain specific (see Section 2). From the analysis done, SSN stands out by far as a well-adopted ontology that follows best practices, and hence serves as the base for FIESTA-IoT ontology. While other IoT related ontologies are bulky and have much processing time, for us, the main aim is to build a solution able to address and maximize interoperability as much as possible.

In order to build FIESTA-IoT ontology, we use various concepts from SSN, IoT-lite, M3-lite taxonomy, DUL and WGS84. We believe that, by using concepts from them, we cover most of the concepts needed towards achieving the goal of interoperability and federation.

#### 3.1 Core Concepts

We have adopted the IoT-A Architecture Reference Model (ARM) [14] core concepts [15] as the foundations to build the ontology. These core concepts are:

- A **Resource** is a “*Computational element that gives access to information about or actuation capabilities on a Physical Entity*”.
- A **Virtual Entity** is a “*Computational or data element representing a Physical Entity*”.

- An IoT **Service** is a “*Software component enabling interaction with IoT resources through a well-defined interface. It can be orchestrated together with non-IoT services (e.g., enterprise services). Interaction with the service is done via the network*”.

In FIESTA-IoT ontology, *Resources* are mainly related to Sensor, Actuator or Tag hosting devices. The conflict between SSN ontology and IoT-A lies in the *Resource* concept. SSN adopts a more device-centric approach. It could be argued that the closest property in SSN that resembles the IoT-A *Resource* is the *Process*. In IoT-A, this property is used for Service sub-model. IoT-A specifies that a *Resource* is hosted on a *Device*, although no information model has been provided for the *Device*. This is where SSN ontology plays an important role in FIESTA-IoT ontology. The *Device* concept is adopted so that a *Resource* is hosted on a `ssn:Device`. This is made explicit using a property e.g. `isHostedOn`. In this case, we redefine the *Resource* concept, although this will not provide any added value to the information, especially upon querying it. Also in IoT-A, multiple *Resources* can be hosted on a single *Device*. However, in SSN, a *Device* can be made up of multiple smaller *Devices*. On this basis, an implicit link (without annotation) between a *Resource* and a *Device* is made, whereby one *Resource* `isHostedOn` one *Device* and hence is treated as one *Entity*. Another aspect to consider is that the *Device* concept in SSN has a subclass that focuses on *Sensing*, i.e. the *SensingDevice*. However, currently SSN only addresses sensing aspects even though IoT-A contemplates other aspects, such as *actuation* and *identification*. Therefore, we instantiate `ssn:Device` for *Actuators* and *Tags* as well. This is where the IoT-lite ontology plays an essential role in extending SSN to include these concepts. In addition, we include the *Virtual Entity* notion (currently missing in SSN and IoT-A ontology) that has been created as an instance of *Entity* in the IoT-lite ontology.

### 3.2 Ontology

As discussed before, the current version of the FIESTA-IoT Ontology<sup>14</sup> is a combination of existing IoT ontologies into a single one with minor updates to address current issues with the existing ontologies. As can be seen from Figure 1, the ontology reuses concepts from a number of “third-party” ontologies and taxonomies such as WGS84, SSN, IoT-lite, M3-lite taxonomy, and DUL. It should be noted that for referencing overlapping concepts in FIESTA-IoT ontology, such as *ActuatingDevice* and *TagDevice*, we chose IoT-lite. For *QuantityKind* and *Units of Measurement*, M3-lite taxonomy that also references QU<sup>15</sup> ontology for various concepts is used while for all other concepts SSN ontology is used.

Summing up the different classes, object and data properties that can be observed in Figure 1, it is worth highlight the following elements:

<sup>14</sup> <https://mimove-apps.paris.inria.fr/ontology/fiestaIoT.html> (also containing the ontology source), also available via <https://github.com/ragarwa2/ontology>

<sup>15</sup> <http://purl.org/NET/ssnx/qu/qu#>

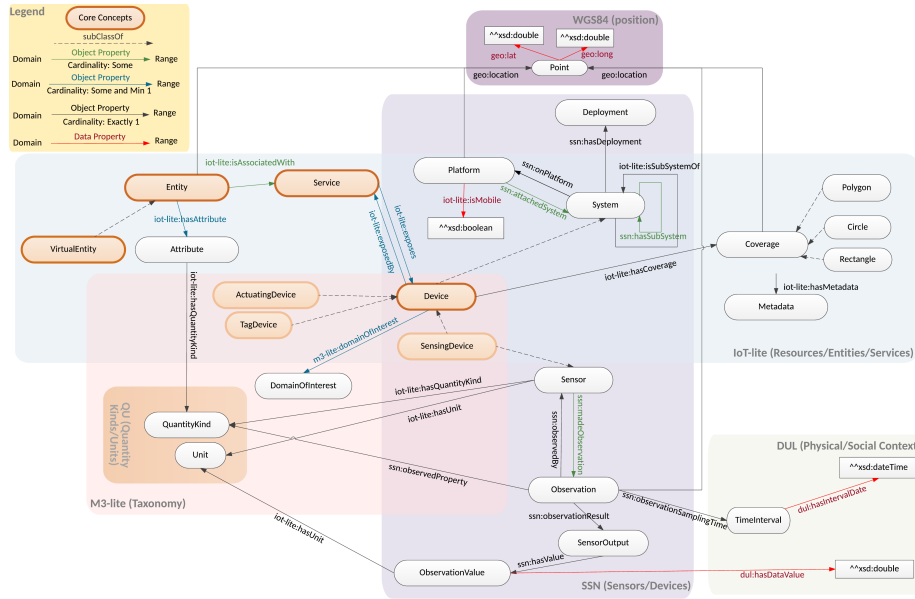


Fig. 1: FIESTA-IoT Ontology.

- `ssn:Deployment` is the root of the graph for every device in order to identify its owner (i.e. testbed).
- `ssn:Platform` is “an Entity to which other Entities can be attached - particularly Sensors and other Platforms. For example, a post might act as the Platform, a buoy might act as a Platform, or a fish might act as a Platform for an attached sensor” [5]. From this description, we attach the physical location of each device. Furthermore, we use the data property `iot-lite:isMobile` to know if the `ssn:Platform` is mobile or not.
- `geo:Point` describes the physical location of the devices and is based on the WGS84 ontology. Within this concept, we use `geo:lat` and `geo:long` data properties to describe the latitude and longitude, respectively (in WGS84 these data properties are described as annotation properties). `geo:Point` is also used to describe the `iot-lite:Coverage` concept and all its underlying subclasses (e.g. Polygon, Circle, Rectangle, etc.). `iot-lite:Coverage` is used in those situations in which a point geographical location is not enough. Using `iot-lite:Coverage` we can define different areas to describe the location (and influence area) of a particular object. As an example, we can define the whole region over which a testbed has a direct influence. In addition, this is also used to describe Virtual Entities for which the `geo:point` location is just not enough.
- `ssn:Device`, core of the resource description, is “a physical piece of technology - a system in a box. Devices may of course be built of smaller devices and software components (i.e. systems have components)” [5]. Within the scope

of our ontology, these devices can be either `iot-lite:ActuatingDevice`, `iot-lite:TagDevice` or `iot-lite:SensingDevice`. From now on we will focus on the latter one.

- `iot-lite:Service` connects to `ssn:Device`. In other words, devices are actually exposed by these IoT services. Note that, since all the types of Devices actually inherit its properties, an IoT service might indistinctly apply for Devices, SensingDevices, etc. Indeed, we actually use the IoT Service endpoint to expose SensingDevices instead of Devices.
- According to the IoT-A principles, Virtual Entities are one of the core parts in an IoT model, together with Devices and Services. By creating a layer upon Devices, Virtual Entities (expressed via `iot-lite:VirtualEntity`) will create associations with a potential number of devices through the observed `iot-lite:Attributes`, which will define the virtual entity properties.
- `iot-lite:SensingDevice` represents physical sensors deployed throughout the different testbeds. As can be seen from Figure 1, `ssn:Sensor` is the superclass of `iot-lite:SensingDevice` and plays an essential role, both in describing different devices and services, and handling the data gathered from the observations.
- `ssn:Sensor` maps to the physical phenomena that it is actually “sensing” via `m3-lite:QuantityKind` and `m3-lite:Units` in the M3-lite taxonomy. More information on the taxonomy is described in 3.3.
- Apart from the physical aspects of `iot-lite:SensingDevice`, it is also important to include other information (i.e. `iot-lite:Metadata`) associated with the observation, to increase its understanding. The `iot-lite:Metadata` can be either the frequency of the measurements, or the accuracy of the sensors, or the precision. Further, we associate `iot-lite:Metadata` to data properties such as `iot-lite:metadataType` and `iot-lite:metadataValue`.
- `m3-lite:DomainOfInterest` represents the domain in which the device is operative.
- Finally, we represent the observations taken by the SensingDevices/Sensors via `dul:TimeInterval`, `geo:Point`, and `ssn:ObservationValue` (this concept links to the actual value of the observation via data property `dul:hasDataValue`, and the corresponding `m3-lite:QuantityKind` and `m3-lite:Unit` concept).

The current concepts in the ontology inherently support streaming data, mobility and reasoning/composite data. The concept `ssn:SensorOutput` can link to multiple `ssn:ObservationValue` via `ssn:hasValue`, thereby providing a mechanism to address streams. Mobility and composite data is handled via `geo:Point`, and `iot-lite:VirtualEntity` and `m3-lite:QuantityKind`, respectively. As each platform has `geo:Point` and all the `ssn:Observation` have `geo:location` object property, mobility of devices with respect to data is also handled. Note, we do not consider handovers and intermittent connectivity in the ontology as our ontology is observation oriented. Next, the composite data or the new kind of data obtained after reasoning or created using `iot-lite:VirtualEntity` is also handled. This requires specific QuantityKind-Unit pair to be available in

Table 1: The M3-lite taxonomy reusing some of the existing IoT ontologies

Prefix	Namespaces
qudt	<a href="http://qudt.org/schema/qudt#">http://qudt.org/schema/qudt#</a>
qu_rec20	<a href="http://purl.org/NET/ssnx/qu/qu-rec20#">http://purl.org/NET/ssnx/qu/qu-rec20#</a>
iot-lite	<a href="http://purl.oclc.org/NET/UNIS/fiware/iot-lite#">http://purl.oclc.org/NET/UNIS/fiware/iot-lite#</a>
owl	<a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>
muo	<a href="http://purl.oclc.org/NET/muo/muo#">http://purl.oclc.org/NET/muo/muo#</a>
rdfs	<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>
ssn	<a href="http://purl.oclc.org/NET/ssnx/ssn#">http://purl.oclc.org/NET/ssnx/ssn#</a>
ucum	<a href="http://idi.fundacionctic.org/muo/ucum-instances.owl">http://idi.fundacionctic.org/muo/ucum-instances.owl</a>
qudt_unit	<a href="http://data.qudt.org/qudt/owl/1.0.0/unit.owl#">http://data.qudt.org/qudt/owl/1.0.0/unit.owl#</a>
spitfire	<a href="http://spitfire-project.eu/ontology/ns/">http://spitfire-project.eu/ontology/ns/</a>
qu	<a href="http://purl.org/NET/ssnx/qu/qu#">http://purl.org/NET/ssnx/qu/qu#</a>
sweet_unit	<a href="http://sweet.jpl.nasa.gov/ontology/units.owl#">http://sweet.jpl.nasa.gov/ontology/units.owl#</a>
openIoT	<a href="http://openiot.eu/ontology/ns/">http://openiot.eu/ontology/ns/</a>
ontoSensor	<a href="http://mmisw.org/ont/univmemphis/sensor">http://mmisw.org/ont/univmemphis/sensor</a>
rdf	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>

M3-lite. To this we envision to have near real-time update to QuantityKind-Unit where users of the ontology can request updates to be performed.

### 3.3 M3-lite Taxonomy

The M3-lite<sup>16</sup> taxonomy is a light version of the M3<sup>17</sup> ontology, tailored to fulfill the FIESTA-IoT ontology main needs and goals. Providing such taxonomy is an essential step since IoT data comes from heterogeneous testbeds using different terms for describing a same phenomenon. The main benefit of the M3-lite taxonomy is to align and interlink numerous already designed IoT-related ontologies to facilitate interoperability (see Table 1 where some of the ontologies reused is mentioned).

Most of the time, the domain ontologies are linked through the `rdfs:SeeAlso` property within M3-lite, such links can be easily ignored if there is no need to deal with such ontologies. M3-lite also follows the idea of “modular ontology” in order to support different needs. The refactoring of the M3 ontology was done to clean non-relevant classes and properties. Further, when the ontologies are reliable (e.g., SSN), instead of using the `owl:equivalentClass` property, M3-lite reuse the concept from the reliable ontology.

The main purpose of the M3-lite taxonomy is to extend SSN, by providing a unified taxonomy for `ssn:Device` concept. In M3-lite, `ssn:SensingDevice` has its own taxonomy and defines 60 top level types of Sensors, `m3-lite:QuantityKind` defines 78 top level physical phenomena, `m3-lite:Unit` defines 64 top level of units of measures, and

<sup>16</sup> <https://mimove-apps.paris.inria.fr/ontology/m3lite.html> (including the ontology source), also available via <https://github.com/ragarwa2/ontology>

<sup>17</sup> <http://sensormeasurement.appspot.com/m3#>



`m3-lite:DomainOfInterest` defines 12 different popular IoT application domains.

### 3.4 Some Stats for the developed Ontology

Currently within the ontology we have: 412 Classes, 30 object properties and 11 data properties in all. Note, we do not consider the data properties of WGS84 as currently they are provided as annotation properties. Further, we have 2 classes that are equivalent (Entity and Object), 7 inverse relations and 1 transitive object. Almost all the object properties and data properties used have domains and ranges specified. Moreover there are many `SubClassOf`, `SubObjectPropertyOf` and `SubDataPropertyOf` relation. In our ontology, the number of class is high because most of them come from M3-lite taxonomy.

### 3.5 Best Practices

When it comes to the creation of an ontology and publishing of the data that use a certain semantic model, best practices should be applied to enhance efficiency, re-usability and interoperability. Below, we list best practices that are followed to create the ontology and guidelines to be followed in order to publish the data.

**Best Practices followed to create unified ontology:** The ontology developed follows Noy et al. methodology [16]. Despite the methodology, the ontology uses concepts from ontologies that already follow best practices (maintained, standardized and recommended by LOV). Further, following steps, also mentioned in [17], are also used:

- Web documentation and tutorial is also provided in order to encourage re-usability. The web documentation is generated using open source ontology documentation tool called `LODE`<sup>18</sup>. The FIESTA-IoT documentation is available online. Note that there are many other tools available (such as `OWLDoc` plugin for `Protégé`, `Parrot`<sup>19</sup> and `WebVOWL`<sup>20</sup>) but we refrain ourselves in describing them as it is out of the scope of this paper.
- Maintenance and online availability of the ontology for easy import.
- Ongoing effort towards standardization of the ontology and cataloging (LOV and LOV4IoT).

**Guidelines for testbeds to publish Data:** Annotated data provides benefits by providing context about the readings or observations that are captured from the real world. But at the same time it adds overhead in terms of communication and storage. Also access to information should be exposed as web-resources, whereby they can be created, accessed, modified and removed. The following steps, also mentioned in [17], should be used as a guide to produce better data:

<sup>18</sup> <http://www.essepuntato.it/lode>

<sup>19</sup> <http://ontorule-project.eu/parrot/parrot>

<sup>20</sup> <http://vowl.visualdataweb.org/webvowl/>

- Registration of Devices and Virtual Entities must identify their respective semantic instances using dereferenceable URIs. This will allow descriptions to be managed and retrieved by using their URIs as URLs.
- Dereferenceable URIs must only be applied to instantiations of `ssn:Device` concept or any of its subclasses (`ssn:SensingDevice`, `iot-lite:ActuatingDevice`, and `iot-lite:TagDevice`).
- Descriptions must be annotated using RDF serialization format such as RDF/XML, JSON-LD, Turtle or N3.
- As the ontology is expected to handle large amounts of data being produced by the testbeds, so the annotation applied to raw/proprietary-formatted data should be minimal, and any triples created must provide added-value towards experimentation. This will alleviate unnecessary load on the testbed when it comes to the delivery and storage.
- Prior to the publication of the annotated data to the data store, the owners of the data are encouraged to validate samples of their descriptions with an ontology validator such as FIESTA-IoT’s Annotation Validator Tool described in section 3.8.

### 3.6 Reference Annotation Tool (RAT)

To provide support for FIESTA-IoT ontology and help testbeds translate from their intrinsic format (e.g. FIWARE-compliant, oneM2M, raw JSON, etc.) to the one that is understood and can be interpreted by others, we provide a sample Reference Annotation Tool (**RAT**) that complies with the ontology created and annotates the desired Device and its observations. The **RAT** will be made public soon via FIESTA-IoT. One of the first steps to be performed when a testbed wants to be a part of a federation, such as FIESTA-IoT, is the registration of all its Devices and IoT Services. The next step is to then annotate the observations produced by the Sensors. The Figures 2 and 3 (along with Figure 4 and 5) show the results provided by the annotator tool that annotates sample Resource/Device and Observations available from the SmartSantander<sup>21</sup> testbed. Here, we focus on the description of a single Device (i.e. `urn:x-iot:smartsantander:u7jcfa:t10000`) and one of the observations generated by one of its Sensor (`urn:x-iot:smartsantander:u7jcfa:t10000.AirTemperature.Sensor`). The annotations provided by the **RAT** follow the guidelines described in Section 3.5. In the Figure 4 and 5, please replace “<platform>” with `urn:x-iot:smartsantander:u7jcfa:t10000`.

### 3.7 Sample Triples and Query

The devices and the data provided by the sensors there in can be stored in the Jena<sup>22</sup> TDB store as it is Semantic data and can be queried using SPARQL<sup>23</sup>

<sup>21</sup> <http://smartsantander.eu/>

<sup>22</sup> <https://jena.apache.org/documentation/tdb/index.html>

<sup>23</sup> <https://www.w3.org/TR/rdf-sparql-query/>

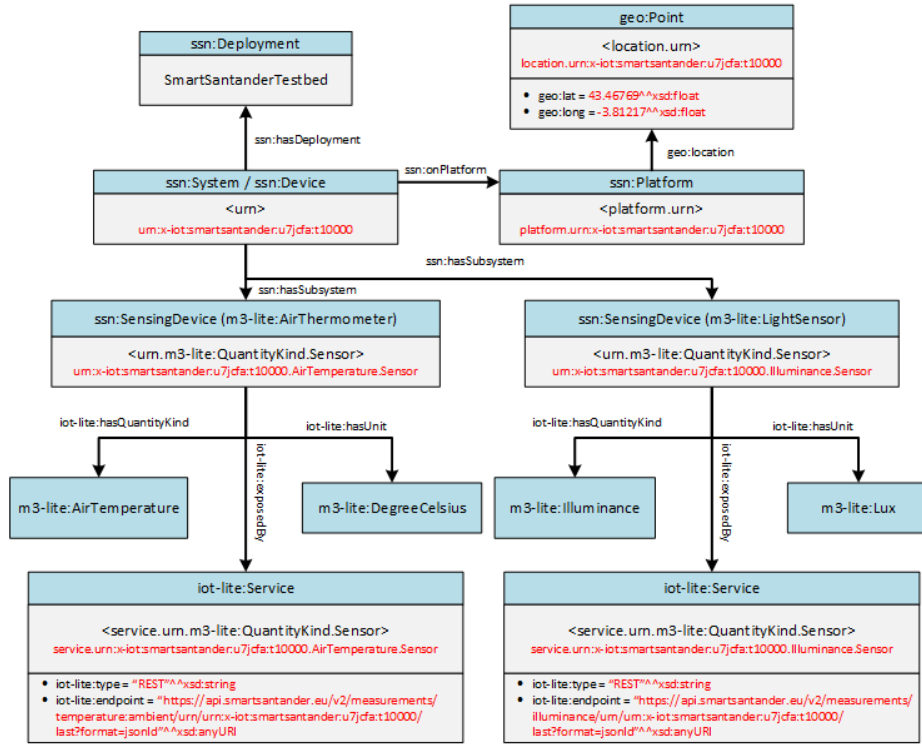


Fig. 2: Annotated Resource Graph.

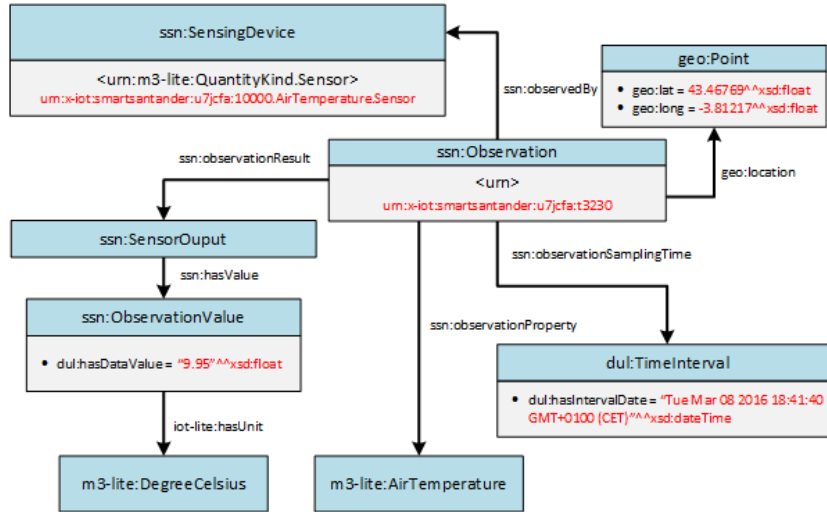


Fig. 3: Annotated Observation Graph.

language. We leverage from Jena store and store the annotated data in the Jena TDB store provided via FIESTA-IoT Meta Cloud Platform. Figure 4 and 5 show sample triples in the turtle format. Note that, these sample triples are generated by the **RAT**. On such data, some sample SPARQL queries that can be executed are as shown in the Figure 6 and 7.

```

1 @prefix owl: <http://www.w3.org/2002/07/owl#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
5 @prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#> .
6 @prefix ssn: <http://purl.oclc.org/NET/ssn/ssn#> .
7 @prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#> .
8 @prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
9 @prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
10 @prefix sms-srd: <http://api.smartsantander.eu#> .
11 @prefix fiesta-iot-srd-href:
12   ↪ <http://platform.fiesta-iot.eu/srd/registry/poc/> .
13 sms-srd:SmartSantanderTestbed a ssn:Deployment .
14 sms-srd:platform.<platform> a ssn:Platform ;
15   geo:location sms-srd:location.<platform> .
16 sms-srd:location.<platform> a geo:Point ;
17   geo:lat "43.47171"^^xsd:float ;
18   geo:long "-3.80014"^^xsd:float .
19 fiesta-iot-srd-href:<platform> a ssn:Device ;
20   ssn:hasDeployment sms-srd:SmartSantanderTestbed ;
21   ssn:hasSubSystem fiesta-iot-srd-href:<platform>.AirTemperature.Sensor;
22   ssn:onPlatform sms-srd:platform.<platform> .
23 fiesta-iot-srd-href:<platform>.AirTemperature.Sensor a ssn:SensingDevice
24   ↪ ;
25 iot-lite:exposedBy sms-srd:service.<platform>.AirTemperature.Sensor ;
26 iot-lite:hasQuantityKind m3-lite:AirTemperature ;
27 iot-lite:hasUnit m3-lite:DegreeCelsius .
28 sms-srd:service.<platform>.AirTemperature.Sensor a iot-lite:Service ;
   iot-lite:endpoint
   ↪ "https://api-dev.smartsantander.eu:10443/v2/measurements/
   ↪ temperature:ambient/urn:<platform>/last?format=jsonld"
   ↪ ^^xsd:string ;
   iot-lite:type "REST"^^xsd:string .

```

Fig. 4: Annotated Resource Triple.

### 3.8 Annotation Validation Tool (AVT)

An important requirement in order to maintain interoperability of information originating from different devices is to make sure that they comply with the

```

1 @prefix owl: <http://www.w3.org/2002/07/owl#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
4 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
5 @prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#> .
6 @prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
7 @prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#> .
8 @prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
9 @prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
10 @prefix fiesta-iot-srd-href:
11   ↪ <http://platform.fiesta-iot.eu/srd/registry/poc/> .
12 _:b0 a ssn:Observation ;
13     ssn:observationResult _:b1 ;
14     ssn:observationSamplingTime _:b2 ;
15     ssn:observedBy fiesta-iot-srd-href:<platform>.AirTemperature.Sensor ;
16     ssn:observedProperty m3-lite:AirTemperature ;
17     geo:location _:b3 .
18 _:b1 a ssn:SensorOutput ;
19     ssn:hasValue _:b4 .
20 _:b2 a dul:TimeInterval ;
21     dul:hasIntervalDate "Fri Apr 01 2016 08:32:34 GMT+0200
22     ↪ (CEST)"^^xsd:dateTime .
23 _:b3 a geo:Point ;
24     geo:lat "-3.80014E0"^^xsd:float ;
25     geo:long "4.347171E1"^^xsd:float .
26 _:b4 a ssn:ObservationValue ;
27     iot-lite:hasUnit m3-lite:DegreeCelsius ;
28     dul:hasDataValue "2.283E1"^^xsd:float .

```

Fig. 5: Annotated Observation Triple.

```

1 PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
2 PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
3 SELECT *
4 WHERE
5   { ?resource a ssn:SensingDevice .
6 }ORDER BY ASC(UCASE(str(?s)))

```

Fig. 6: SPARQL Query to get all information about the device of type `ssn:SensingDevice`.

```

1 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
2 PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
3 PREFIX ssn: <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#>
4 PREFIX m3-lite:<http://purl.org/iot/vocab/m3-lite#>
5 SELECT ?s ?lat ?lng ?endp
6 WHERE
7   { ?s a ssn:SensingDevice .
8     ?s iot-lite:isExposedBy ?serv .
9     ?s iot-lite:hasQuantityKind m3-lite:AirTemperature .
10    ?s ssn:onPlatform ?platform .
11    ?platform geo:location ?point .
12    ?point geo:lat ?lat .
13    ?point geo:long ?lng .
14    ?serv iot-lite:endpoint ?endp.
15    FILTER (?lng > -50 && ?lng < 50 && ?lat < 100 && ?lat > 0 )
16  }

```

Fig. 7: SPARQL Query to get devices that are of type `ssn:SensingDevice` and are based in certain area characterized by their geographical location.

presented unified semantic model. Therefore, a validation check is required for the compliance of the annotations. The validation should check for syntactic and semantic issues. Among these issues, some syntactic issues are: *(i)* unknown properties and classes with respect to the unified ontology, *(ii)* problematic prefix namespaces, *(iii)* ill-formed URIs and language tags on literals, *(iv)* data-typed literals with illegal lexical forms, *(v)* unexpected local names in schema namespaces, *(vi)* untyped resources and literals, and *(vii)* individuals having consistent types, assuming complete typing broken RDF list structures. While semantic inconsistencies are: *(i)* inheritance relationships for classes and properties, *(ii)* Cardinality, and *(iii)* unexpected domains and ranges.

Such validation should be performed at the platform performing the federation when the data is registered. As an ongoing effort towards **AVT**, we are trying to adopt the SSN validator<sup>24</sup>, apply necessary changes, and make it available via FIESTA-IoT. In **AVT**, both syntactic and semantic validation will be applied instead of only syntactic validation as currently available with SSN validator. Further, an evaluation of delays introduced by the process will also be done.

## 4 Current Implementations and uses of the Ontology

As FIESTA-IoT ontology is data oriented, currently it finds its implementation in the EU funded H2020 FIESTA-IoT project that aims to provide federation and interoperability to the IoT devices and the data produced by them in an

<sup>24</sup> <http://iot.ee.surrey.ac.uk/SSNValidation/>

agnostic way. However, the ontology is just not specific to FIESTA-IoT platform. In our vision, the ontology can also be applied to all IoT projects willing to semantically annotate data that is produced by devices and store locally the measurements. We also see the applicability of our ontology to testbed that comply either to FIRE<sup>25</sup> (such as FED4FIRE<sup>26</sup> or FI-WARE<sup>27</sup>) or to GENI<sup>28</sup> or to oneM2M standardization or even to other Future Internet initiatives. Currently we see SmartSantander, Com4Innov, SmartICS and Mobius testbeds (whose actual owners are part of the FIESTA-IoT consortium) as prime users of both FIESTA-IoT ontology, RAT and AVT.

## 5 Conclusion

In this work we present a unified ontology (FIESTA-IoT ontology) that aims at addressing interoperability issues. The motivation to build the such unified ontology comes from: *(i)* not overloading the domain with a new ontology but integrating various existing required ontologies (i.e. the needed concepts) into a single and holistic one in order to fulfill the needs of the testbeds, *(ii)* reusing as much possible the existing ontologies in order to help testbeds not re-annotate their dataset in order to join the federation, *(iii)* ensuring a better interoperability with existing semantics-based IoT platforms, projects, and standardizations, and *(iv)* following best practices. The FIESTA-IoT ontology and M3-lite are already submitted to LOV and LOV4IoT.

Nevertheless, the current version of the ontology lacks integration with some existing standardization such as oneM2M. We are currently working to address such issues and would like to address them in the second version of the unified ontology (envisioning that nothing much will change). Further, there is an ongoing effort towards evaluating our proposed ontology with the mechanism described in [18]. We also envisage to update M3-lite taxonomy by other domain knowledge and provide a mechanism for suggesting updates to the taxonomy.

## 6 Acknowledgment

This work is partially funded by the European project “Federated Interoperable Semantic IoT/cloud Testbeds and Applications (FIESTA-IoT)” from the European Union’s Horizon 2020 Programme with the Grant Agreement No. CNECT-ICT-643943. The Authors would like to also thank the FIESTA-IoT consortium for fruitful discussions.

<sup>25</sup> <https://www.ict-fire.eu>

<sup>26</sup> <http://www.fed4fire.eu/fire>

<sup>27</sup> <https://www.fiware.org>

<sup>28</sup> <https://www.geni.net>

## References

1. Gavras, A.: Experimentally Driven Research White Paper. Technical report, ICT-Fireworks (2010)
2. IEEE: IEEE Standard Glossary of Software Engineering Terminology (1990)
3. Ye, J., Coyle, L., Dobson, S., Nixon, P.: Ontology-based models in pervasive computing systems. *Knowl. Eng. Rev.* **22**(4) (December 2007) 315–347
4. Compton, M., Henson, C., Lefort, L., Neuhaus, H., Sheth, A.: A Survey of the Semantic Specification of Sensors. In: *Proc. of 2nd International workshop of Semantic Sensor Networks*, Washington DC (oct 2009) 17–32
5. Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web* **17** (dec 2012) 25–32
6. IoT-A: Deliverable D4.3: Concepts and solutions for Entity-Based Discovery of IoT Resources and Managing their Dynamic Association. Technical report (2012)
7. Leggieri, M., von der Weth, C., Serrano, M.: Deliverable 3.1.2: Semantic Representations of Internet-Connected Objects. Technical report (2013)
8. Hachem, S., Teixeira, T., Issarny, V.: Ontologies for the Internet of Things. In: *Proceedings of the 8th Middleware Doctoral Symposium on - MDS '11*, Lisbon, ACM Press (dec 2011) 1–6
9. Seydoux, N., Alaya, M.B., Hernandez, N., Monteil, T., Haemmerle, O.: Semantique et Internet des objets : dun etat de lart a une ontologie modulaire. In: *26es Journees francophones dIngenierie des Connaissances*, Rennes (2015)
10. OneM2M: TS-0012 oneM2M Base Ontology. Technical report (2016)
11. Gyrard, A., Datta, S.K., Bonnet, C., Boudaoud, K.: Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation. In: *3rd International Conference on Future Internet of Things and Cloud*, Rome, IEEE (Aug 2015) 9–16
12. Daniele, L., den Hartog, F., Roes, J.: Study on Semantic Assets for Smart Appliances Interoperability. Technical report, EU (2015)
13. Willner, A., Papagianni, C., Giatili, M., Grosso, P., Morsey, M., Al-Hazmi, Y., Baldin, I.: The Open-Multinet Upper Ontology Towards the Semantic-based Management of Federated Infrastructures. In: *10th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM)*, Vancouver, Canada, ACM (aug 2015) 10
14. IoT-A: Deliverable D1.5: Final Architectural Reference Model for the IoT. Technical report, IoT-A (2013)
15. IoT-A: IoT-A Terminology
16. Noy, N.F., McGuinness, D.L.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Technical report, Knowledge Systems Laboratory, Stanford (2001)
17. Gyrard, A., Serrano, M., Atemezing, G.A.: Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In: *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, IEEE (dec 2015) 412–417
18. Duque-Ramos, A., Boeker, M., Jansen, L., Schulz, S., Iniesta, M., Fernandez-Breis, J.T.: Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuARE). *PLoS ONE* **9**(8) (Aug 2014) e104463 (1–14)