HORIZONS 2020 PROGRAMME

Research and Innovation Action – FIRE Initiative

Call Identifier:	H2020–ICT–2014–1
Project Number:	643943
Project Acronym:	FIESTA-IoT
Project Title:	Federated Interoperable Semantic IoT/cloud Testbeds and Applications

Semantic Models for Testbeds, Interoperability and Mobility Support, and Best Practices V1

Document Id:	FIESTAIoT-D31-160515-Draft
File Name:	FIESTAIoT-D31-160515-Draft.pdf
Document reference:	Deliverable 3.1
Version:	Draft
Editor:	Rachit Agarwal, Nikolaos Georgantas, Valerie Issarny
Organisation:	Inria
Date:	05 / 15 / 2016
Document type:	Deliverable
Dissemination level:	PU

DOCUMENT HISTORY

Rev.	Author(s)	Organisation(s)	Date	Comments
V01	Martin Serrano	NUIG-DERI	2015/02/01	Initial Draft Proposal
	Amelie Gyrard	NUIG-DERI	2015/07/24	Brainstorming document created, Aligning IoT ontologies, State of the art ontologies
	Maria Bermudez-Edo	UNIS	2015/07/28	Initial Draft Proposal
	Rachit Agarwal	Inria	2016/01/04	ToC defined and first approach
	Amelie Gyrard	NUIG-DERI	2016/02/03 2016/03/03 2016/03/07 2016/03/09 2016/03/11 2016/03/14 2016/03/16	SSN ontology section M3-lite & VITAL sections Added a point in the Future work section SAOPY, Sense2Web M3 ontology section, update m3-lite pictures, M3 ontology Description of ontologies (IoT-O ontology), methodologies description (Ontology methodologies, Ontology modularization, Ontology design patterns) Table ontology namespace at the beginning of the document, fix issues references, update table acronyms
	Nikos Kefalakis	AIT	2016/03/14	OpenIoT Ontology
	Rachit Agarwal	Inria	2016/03/14 2016/03/15 2016/03/16 2016/03/17 2016/03/18 2016/03/21 2016/03/22 2016/03/23 2016/03/31	Version available for collaborative editing Merge contributions from AIT Update related work section Add content to mobility section Added SAREF ontology (Extracts from Mengxuan Zhao) Updates section mobility section Update positioning Section Update Ontology namespaces table Update section FIESTA-IoT Ontology and mobility sections Update mobility section, manage References by properly aligning them to a format
	Tarek Elsaleh	UNIS	2016/03/13 2016/03/24 2016/03/31 2016/04/04	Update related work, FIESTA-IoT ontology, and Annotation section Update related work section Update FIESTA-IoT ontology and Tools section Update validation tools section
V02	David Gomez Luis Sanchez Jorge Lanza	UC	2016/03/30 2016/04/04-07 2016/04/08	Update related work section Update related work section Update FIESTA-IoT ontology and annotation tools section Add mapping to M3-lite taxonomy
	Roland Steinke Alex Willner	FOKUS	2016/03/31	Updated Open-Multinet ontology
	Rachit Agarwal	Inria	2016/04/01 2016/04/04	Update ontology aligned section Create Appendix for QK and units Add Appendices for Documentation of M3-lite and FIESTA-IoT, and sample annotations, add composition and data/services section

			2016/04/08 2016/04/12 2016/04/13 2016/04/14 2016/04/15	Merge updates from Alex Willner Add importance of ontologies in Related work Update FIESTA-IoT Ontology, add section for sample queries Future work and executive summary Documentation and OWL version of FIESTA-IoT Ontology, review text for all the sections
V02	Minwoo Ryu	KETI	2016/04/06 2016/04/11	Updated oneM2M Base Ontology Added information of oneM2M Base Ontology
	Amelie Gyrard	NUIG-DERI	2016/04/08 2016/04/14	LOV4IoT section, move appendix for documentation to main section and add text about LODE, Protégé, WebVOWL, Parrot. Ontology networks vision to interconnect domain ontologies, service composition vision based on semantic web services, other updates
	Tarek Elsaleh	UNIS	2016/04/13 2016/04/14	Update related work section, update Guidelines and best practices section Update FIESTA-IoT validation tools
V03	Rachit Agarwal	Inria	2016/04/15 2016/05/03	Version ready for reviewing Request for Assignment of Reviewers sent Reviewers assigned by Rachit
	Juan Echevarría	SDR	2016/05/05	Quality Review of V03
	Rachit Agarwal	Inria	2016/05/09	Quality review integrated
V04	Mengxuan Zhao	EGM	2016/05/09	Technical Review 1
	Roland Steinke	FOKUS	2016/05/10	Technical Review 2
V05	David Gomez	UC	2016/05/11	Reply to Technical Review for UC sections
	Amelie Gyrard	NUIG	2016/05/13	Reply to Technical Review for NUIG sections
V06	Rachit Agarwal	Inria	2016/05/13	Generation of Final version
V07	Martin Serrano	NUIG	2016/05/15	Circulated for Approval
Draft	Martin Serrano	NUIG	2016/05/15	EC Submitted

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMS AND ACRONYMS

Acronym	Definition
API	Application Programming Interface
ARM	Architecture Reference Model
CDMI	Cloud Data Management Interface
CESN	Coastal Environment Sensor Network
CoAP	Constrained Application Protocol
DSL	Domain Specific Language
DUL	Dolce+DnS Ultralite
EaaS	Experimentation-as-a-Service
ETSI	European Telecommunications Standards Institute
EU	European Union
FIRE	Future Internet Research and Experimentation
FP	Framework Programme
GENI	Global Environment for Network Innovations
GSN	Global Sensor Network
ICO	Internet Connected Object
IEEE	Institute of Electrical and Electronics Engineers
INDL	Infrastructure and Network Description Language
IoT	Internet of Things
IoT-A	Internet of Things Architecture
IoT-O	Internet of Things Ontology
IP	Internet Protocol
IRI	Internationalized Resource Identifier
IT	Information Technology
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
KM4City	Knowledge Model for City
KPI	Key Performance Indicators
LOD	Linked Open Data
LODE	Live OWL documentation Environment
LOV	Linked Open Vocabularies
LOV4IoT	Linked Open Vocabularies for Internet of Things
M2M	Machine-to-Machine
M3	Machine-to-Machine Measurement
MAS	Management, Abstraction and Semantics
MSM	Minimal Service Model
N3	Notation 3
NeOn	Network of Ontology

NGSI	Next Generation Service Interface
NML	Network Mark-Up Language
NOVI	Networking innovations Over Virtualized Infrastructures
OCCI	Open Cloud Computing Interface
ODPs	Ontology Design Patterns (ODPs)
OMN	Open-Multinet
oneM2M	International Machine-to-Machine Standardization
OTN	Ontology Transportation Networks
OWL	Web Ontology Language
QK	QuantityKind
QoS	Quality of Service
RDF	Resource Description Format
RFID	Radio Frequency Identification
S-LOR	Sensor-based Linked Open Rules
SAN	Semantic Actuator Network
SAO	Stream Annotation Ontology
SAREF	Smart Appliances REFerence
SenML	Sensor Markup Language
SI	International System of Units
SLA	Service Level Agreements
SSH	Secure Shell
SSN	Semantic Sensor Networks
SWRL	Semantic Web Rule Language
TTL	Time to Live
VE	Virtual Entity
W3C	World Wide Web Consortium
WebVOWL	Web Visual Notation for OWL Ontologies
WP	Work Package
WSN	Wireless sensor Network
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

Ontology Namespaces

Prefix	Ontology/Language	Namespace
aws	Ontology for Meteorological sensors	http://purl.oclc.org/NET/ssnx/meteo/aws
dcn	Delivery Context Ontology	http://www.w3.org/2007uwa/context/deliveryContext.owl#
dcterms	DCMI Metadata Terms	http://purl.org/dc/terms/
dul	DOLCE+DnS Ultralite Ontology	http://www.loa.istc.cnr.it/ontologies/DUL.owl#
es	Example	http://www.example.org/ns#
foaf	Friend of a Friend	http://xmlns.com/foaf/
geo	Basic Geo (WGS84) ontology/ WGS84 Geo Positioning	http://www.w3.org/2003/01/geo/wgs84_pos#
goodRelations	GoodRelations	http://purl.org/goodrelations/v1
hachem	Ontology by Hachem et al.	• http://websvn.ow2.org/filedetails.php?repname=choreos&path=%2Ftrunk%2Fextensible-service-discovery%2Fregistration_manager%2FsensorsAndActuators.rdf • http://websvn.ow2.org/filedetails.php?repname=choreos&path=%2Ftrunk%2Fextensible-service-discovery%2Fquery_manager%2FsensorsAndActuatorsComp.rdf • http://websvn.ow2.org/filedetails.php?repname=choreos&path=%2Ftrunk%2Fextensible-service-discovery%2Fquery_manager%2FunitConversion.rdf
hrest	hRESTS ontology	http://www.wsmo.org/ns/hrests#
iot-a	IoT-A Ontology	• http://www.surrey.ac.uk/ccsr/ontologies/ResourceModel.owl# • http://www.surrey.ac.uk/ccsr/ontologies/VirtualEntityModel.owl# • http://www.surrey.ac.uk/ccsr/ontologies/ServiceModel.owl#
iot-lite	IoT-lite	http://purl.oclc.org/NET/UNIS/fiware/iot-lite#
iot-O	IoT Ontology	https://www.irit.fr/recherches/MELODI/ontologies/IoT-O
m3	M3	http://sensormeasurement.appspot.com/m3#
m3-lite	M3-lite	http://purl.org/iot/vocab/m3-lite#
msm	Minimal Service Model ontology	http://iserve.kmi.open.ac.uk/ns/msm#
muo	Measurement Unit Ontology	http://purl.oclc.org/NET/muo/muo#
oneM2M	oneM2M base Ontology	http://www.onem2m.org/ontology/Base_Ontology#
onm	Open-Multinet	http://open-multinet.info#
openiot	OpenIot	http://www.openiot.eu/ontology/ns
otn	Ontology of Transportation Network	http://www.pms.ifi.lmu.de/rewerse-wgal/otn/OTN.owl
owl	Web ontology Language	http://www.w3.org/2002/07/owl#

prov	PROVO ontology	http://www.w3.org/ns/prov#
qoi	Quality of Information	http://purl.oclc.org/NET/UASO/qoi#
qu-rec20	QuantityKind	http://purl.org/NET/ssnx/qu/qu-rec20#
qudt	Quantities, Units, Dimensions and Data Types Ontologies	http://qudt.org/schema/qudt#
qudt_unit	QUDT unit ontology	http://data.qudt.org/qudt/owl/1.0.0/unit.owl#
rdf	RDF Concepts Vocabulary	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	RDF Schema ontology	http://www.w3.org/2000/01/rdf-schema#
s4ac	Social Semantic SPARQL Security for Access Control	http://ns.inria.fr/s4ac/v2#
san	Semantic Actuator Network ontology	https://www.irit.fr/recherches/MELODI/ontologies/SAN
sao	Stream Annotation Ontology	http://purl.oclc.org/NET/UNIS/sao/sao#
saref	Smart Appliances REFerence Ontology	http://ontology.tno.nl/saref.ttl
sawdl	Semantic Annotations for WSDL and XML Schema ontology	http://www.w3.org/ns/sawsdl#
shw	Smart Home Weather	http://paul.staroch.name/thesis/SmartHomeWeather.owl#
spt	SPITFIRE ontology	http://spitfire−project.eu/ontology/ns/
ssn	W3C SSN ontology	http://purl.oclc.org/NET/ssnx/ssn#
sweet	Semantic Web for Earth and Environmental Terminology	https://sweet.jpl.nasa.gov
sweet_unit	SWEET unit ontology	http://sweet.jpl.nasa.gov/ontology/units.owl#
time	OWL time ontology	https://www.w3.org/TR/owl-time/
timeline	Timeline ontology	http://purl.org/NET/c4dm/timeline.owl#
vital	VITAL ontology	http://vital-iot.com/ontology#
wgs84_pos	Basic Geo (WGS84 lat/long) Vocabulary	http://www.w3.org/2003/01/geo/wgs84_pos#
wsl	WSMO-Lite ontology	http://www.wsmo.org/ns/wsmo-lite#
xsd	XML schema Definition	http://www.w3.org/2001/XMLSchema#

1 POSITIONING

1.1 FIESTA-IoT Scope

Recent advances in the Internet of Things (IoT) area have progressively moved in different directions (i.e. designing technology, deploying the systems into the cloud, increasing the number of inter-connected entities, improving the collection of information in real-time, and no less important—the security aspects in IoT. IoT advances have drawn a common grand challenge that focuses on the integration of the heterogeneous IoT generated data. This key challenge is to provide a common sharing model or a set of models organizing the information coming from the connected IoT services, IoT technology and systems, and more important, to be able to offer them as experimental services in order to optimize the design of new IoT systems and facilitate the generation of solutions more rapidly.

In FIESTA-IoT we focus on the problem of formulating and managing IoT data from heterogeneous systems and environments and their entity resources (such as smart devices, sensors, actuators, etc.), this vision of integrating IoT platforms, Testbeds and their associated silo applications within cloud infrastructures is related to several scientific challenges, such as the need to aggregate and ensure the interoperability of data streams stemming from different IoT platforms or Testbeds, as well as the need to provide tools and techniques for building applications that horizontally integrate diverse IoT Solutions. The convergence of IoT with cloud computing is a key enabler for this integration and interoperability, since it allows the aggregation of multiple IoT data streams towards the development and deployment of scalable, elastic and reliable applications that are delivered on-demand according to a pay-as-you-go model.

The activity in FIESTA-IoT is distributed in 7 Work Packages (WP). WP1 is dedicated to the project activities coordination, considering consortium administration, financial management, activity co-ordination, reporting and quality control. In FIESTA-IoT one of the main objectives is to include experimenters and new Testbeds to test and provide feedback about the platform and tools, thus open calls for those tenders will be issued (these are also part of the WP1 activity and it is called selection of third-parties).

WP2 focuses on stakeholder’s requirements and the analysis of IoT platforms and Testbeds in order to define strategies for the definition and inclusion of experiments, tools and Key Performance Indicators (KPIs). The activities in WP2 are focused on studying the IoT platforms and Testbeds and the specification of the experiments, the detail of the needed tools for experimentation, and the KPIs for validating the proposed solutions. This WP will conduct the design and development of the Meta-Cloud Architecture (including the relevant directory of IoT resources) and will define the technical specification of the project. WP2 also focuses on analysing the Global Market Confidence program and establishes the Certification Program Specifications that will drive the global market confidence and certification actions around the IoT experimentation model.

WP3 focuses on providing technologies, interfaces, methods and solutions to represent the device and network nodes of the Testbeds as virtualized resources. The virtualized resources will be represented as services and will be accessible via common service interfaces and Application Program Interfaces - APIs (i.e. the FIESTA-IoT Testbed interfaces/APIs). The virtualized resources and their capabilities and interfaces will be also described using semantic metadata to enable (semi-) automated discovery, selection and access to the Testbed devices and resources.

WP4 will implement an infrastructure for accessing data and services from multiple distributed diverse Testbeds in a secure and Testbed agnostic way. To this end, it will rely on the semantic interoperability of the various Testbeds (realized in WP3) and implement a single entry point for accessing the FIESTA-IoT data and resources in a seamless way and according to an on-demand Experimentation-as-a-Service (EaaS) model. The infrastructure to be implemented will be deployed in a cloud environment and will be accessible through a unified portal infrastructure.

WP5 focuses on designing, deploying and delivering a set of experiments, so as to assess the feasibility and applicability of the integration and federation techniques, procedures and functions developed during the project lifetime. It will define a complete set of experiments to test the developments coming from other WPs (mainly WP3 and WP4), covering all of the specifications and requirements of WP2. Developments will be tested over available IoT environments and/or smart cities platforms. WP5 will also provide evaluation of the KPIs defined for every experiment/pilot. The final deployed experiments will include a subset of those coming from WP2, WP3 and WP4, as well as those provided by FIESTA-IoT Open Calls.

WP6 focuses on the establishment and validation of the project’s global market confidence on IoT interoperability, which will provide a vehicle for the sustainability and wider use of the project’s results. The main activity in this WP focuses on specifying and designing an IoT interoperability program, including a set of well-defined processes that will facilitate the participation of researchers and enterprises. WP6 works on providing a range of certification and compliance tools, aimed at auditing and ensuring the openness and interoperability of IoT platforms and technologies. WP6 also focuses on interoperability testing and validation and to provide training, consulting and support services to the FIESTA-IoT participants in order to facilitate platforms and tool usability, but also to maximize the value offered to them by using FIESTA-IoT suite and tools.

WP7 focuses on ensuring that the FIESTA-IoT suite, models and tools engage well with the community outside of the project; from promotion and engagement of new customers, to the front line support of current users, and the long-term exploitation of results and sustainability of the facility itself. This will be carried out in a coordinated manner such that a consistent message and professional service is maintained. Dissemination activities and the KPI to measure the impacts will be studied and used in this WP. An ecosystem plan including the specification of processes, responsibilities and targets will be generated and the evaluation and effectiveness of the operating model will be evaluated within this WP. In this WP the successes of stakeholder engagement and reports on their satisfaction with the services offered in FIESTA-IoT will be put in place at the end of the project.

1.2 WP3 overview

This work package focuses on providing technologies, interfaces, methods and solutions to represent the device and network nodes of the test-beds as virtualized resources. The virtualized resources will be represented as services and will be accessible via common service interfaces and APIs (i.e. the FIESTA-IoT Testbed interfaces/APIs). The virtualized resources and their capabilities and interfaces will be also described using semantic metadata to enable (semi-) automated discovery, selection and access to the Testbed devices and resources. The virtualized resources will enable access to the devices for management and configuration proposes, will allow sending actuation commands and feedback to the test-bed devices and will also provide interfaces to the streaming data emerging from the resources. The data streams will represent the observation and

measurement data that is collected by the Testbed resources. With respect to FIESTA-IoT objectives, WP3 main objectives are to:

- Provide an higher-level service abstraction and virtualized representations of the Testbed resources;
- Enable access and configuration of the Testbed resources using common interfaces;
- Develop (semantic) annotation models for interoperable data/service exchange between the Testbeds and also between Testbeds and higher-level services/applications;
- Enable (Semantic) Stream reasoning, provide re-usable components and develop common methods for data analytics for data streams;
- Provide common interfaces, and command and control mechanisms for real-time management and configuration of the streams.

The activities in this work package first start with developing common models and (semantic) annotation frameworks to describe the Testbed resources. The activities will also focus on providing common interfaces to access the Testbed resources, represent their streaming data and exchange configuration, control and actuation commands with the Testbed resources. Developing and/or extending common description models for the services and the IoT data provide semantic interoperability for the services and the data streams. The work package activities also provide re-usable software components, tools and mechanisms to process the streaming IoT data and to manage the stream using real-time or near real-time control and monitoring mechanisms.

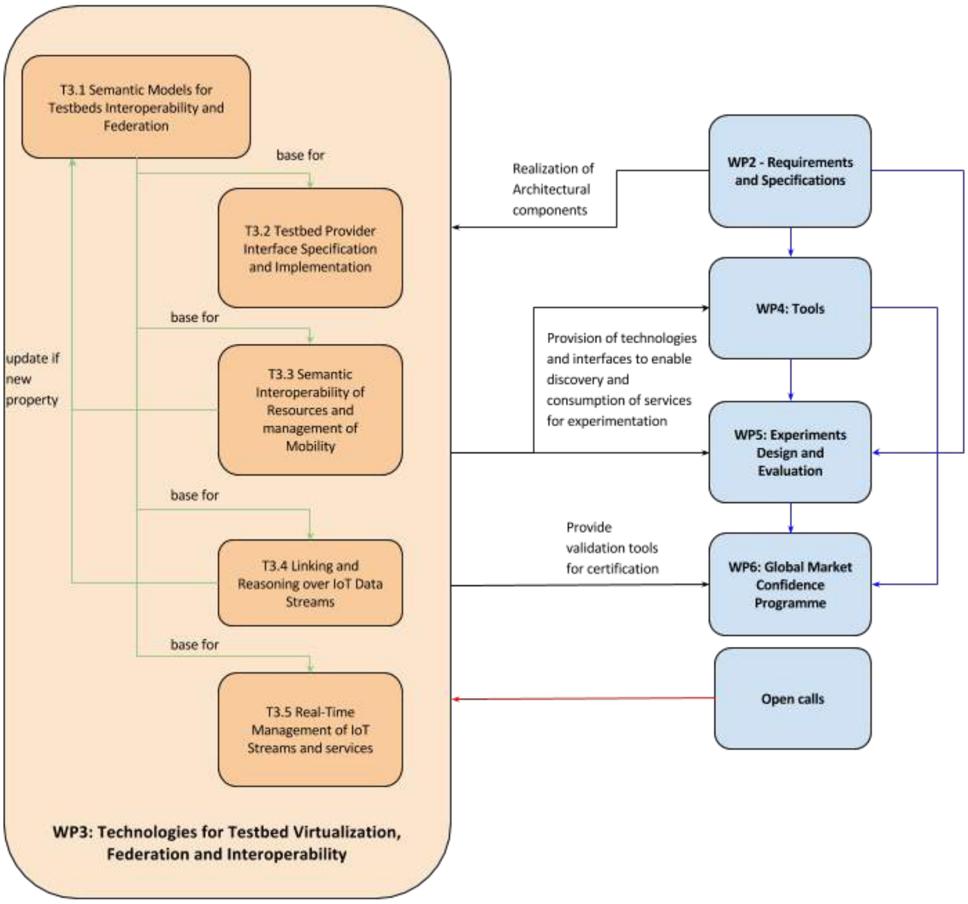


Figure 1: WP3 relation with different work packages

The WP3 Tasks cross all aspects of the FIESTA-IoT Infrastructure and links with different work packages (See Figure 1). The set of Tasks (T) within WP3 are:

- **T3.1 Semantic Models for Testbeds Interoperability and Federation**: This task is aimed to develop common semantic description models to describe the Testbed resources, services and the IoT data. There are several existing models for describing the IoT data, services and network resources. Some of the existing Testbeds have also adapted various description models and interfaces for their Testbeds. The activities in this context mainly focus on re-using the existing common models, extending and/or adapting the common description frameworks and providing optimum, customizable, extensible and modular semantic description framework to enable interoperability among different Testbeds and data/service providers. The semantic models are provided for Testbed devices, IoT data and data streams. A set of tools and (semi-automated) mechanisms will be developed to enable federation of resource and creation of composite data/services using multiple resources. The semantic description model developed in this activity is used in T3.3 for interoperability test and evaluation.

- **T3.2 FIESTA-IoT Testbed Provider Interfaces Specification and Implementation**: This task is devoted to develop common interfaces in order to access and exchange data and command among the Testbed resources and higher-level services and applications. For the interface development, existing common models and enablers such as Next Generation Service Interfaces (NGSI) and the services models that are developed in the FP7 IoT-A project as well as the common interfaces that are currently been used in the existing Testbeds will be investigated. This activity will then provide a set of common interface specification with a focus on compatibility with the most common technologies (e.g. RESTful, supporting technologies such as Constrained Application Protocol (CoAP)) and existing interfaces (e.g., NGSI, Internet of Things-Architecture (IoT-A)) and will also provide a reference implementation of those interfaces on the Testbeds. Furthermore, the adaptation of standards-based cloud interfaces (such as Open Cloud Computing Interface (OCCI), SNIA's Cloud Data Management Interface (CDMI), DeltaCloud APIs) will be enhanced, adapted and used for accessing IoT clouds. Using common interfaces in collaboration with the common semantic description models in T3.1 will enable interoperability and flexible integration of the resources among heterogeneous Testbeds.

- **T3.3 Semantic Interoperability of Resources and Management of Mobility**: This task will provide tools, interfaces and will describe best practices to publish data and services according to the semantic models that will be developed in this activity. A set of tools and interfaces will be also provided to enable a third-party developer to test and evaluate their semantic description against the provided models and to verify their semantic interoperability and/or receive feedback to enhance and adapt their models. We will also provide mechanisms for different serialization formats and will also evaluate the efficiency, size and complexity of various representations according to different use-case scenarios. The mobility of the resources, updating the description of the pervasive resources and supporting access and subscription to the mobile resources via the Testbeds will be also investigated in this activity. Mechanisms to enable management of the mobile resources will be provided to enable access and hand-over between different gateway/Testbed resources in mobile scenarios.

- **T3.4 Linking and Reasoning over IoT Data Streams**: This task will focus on data analytics and developing mechanisms to process the real-time dynamic IoT data streams. The data analytics method will initially focus on stream processing and creating patterns, data abstractions from real-time data that emerges from the test-bed as well as processing the static and historical data. We will investigate the association and reasoning methods and will develop solutions to detect different correlations between the patterns in the streams and also multiple and multi-modal data streams. The multi-modality of data and linking between different data streams will be a key challenge to solve and for this purpose we will use knowledge-based solutions (that will also utilize the semantic descriptions provided in T3.1) and reasoning mechanisms as well and machine learning techniques to process and interpret the data. The information extraction techniques will provide actionable knowledge that can be used by higher-level services and applications. The main focus will be on providing the developed methods and techniques as re-usable and common components with a set of common interfaces that can be used over various Testbed and virtualized data streams and can be also called by different higher-level services and applications.

- **T3.5 Real-Time Management of IoT Streams and Services**: This task will develop a set of mechanisms and interfaces to manage the IoT streams and to submit/receive the control and feedback commands to/from the lower-level resources that provide the IoT data. The control and feedback command and management mechanisms will enable adapting and changing the data access, publication and in-network processing parameters in (near) real-time by considering the device level and Testbed level resources such as changing the data collection and communication frequency when a device has low battery, providing in-network abstraction, aggregation to decease the emerging data traffic, changing the sampling frequency or setting neighbouring devices to on/off based on the resolution that is required. These activities will require a set of management components and common command and feedback mechanisms and also intelligent learning and adaptation methods to enable automated and real-time control of the IoT devices and services.

Table 1: WP3 Deliverables

No.	Deliverable	Related Task	Responsible Partner	Contributors
D3.1	Semantic models for Testbeds, interoperability and mobility support, and best practices	T3.1 T3.3	Inria, UNIS	NUIG-DERI, AIT, KETI, UNICAN, FOCUS
D3.2	Specification and implementation of common Testbed interfaces	T3.2	AIT	UNICAN, UNIS, NUIG-DERI, Com4Innov
D3.3	Concept and Development for IoT data analytics and IoT stream and service management	T3.4 T3.5	NUIG-DERI, UNIS	UNICAN, KETI, Inria

1.3 Audience

This deliverable addresses following audiences:

- **Researchers and engineers within the FIESTA-IoT consortium** will take into account various requirements in order to research, design and implement the APIs needed to support Testbeds associated to FIESTA-IoT Platform.
- **Testbed owners who wish to join FIESTA-IoT** will be able to use the tools to annotate the data their Testbed is producing. These annotation tools should comply with the semantic model proposed within FIESTA-IoT. By doing so, the Testbed can either become Class I, Class II, or Class III Testbed (see [1] for the definitions of various classes of Testbeds).
- **Experiment owners who wish to join FIESTA-IoT** will be able to understand how and what IoT data is stored within FIESTA-IoT Meta Cloud and thus would be able to align their experiments that could utilize such data.
- **Researchers on Future Internet Research and Experimentation (FIRE) focusing on semantically storing data produced by their experiments** will find guidelines to store data produced by their experiments in a semantic manner either in their own repository or utilizing FIESTA-IoT platform. The researchers will be able use the ontology and the tools as the reference. Further, if they wish to extend/modify the ontology and tools for their own research, they would be able to do so.
- **Members of other Internet of Things (IoT) communities and projects (such as projects of the IERC cluster)** can take this document as an initial reference or inspiration to design and implement their own Testbed that also stores data that is semantically annotated.
- **Open call** participants will be able to understand better the technical details needed for them to join the FIESTA-IoT consortium.
- **Standardization bodies** will have access to this deliverable as it will be a public document and therefore the ontology developed can be standardized following the involvement and reach a wider adoption.

1.4 Terminology and Definition

This sub-section is intended to clarify the terminology used during this project. This initial step is intended to clarify all of the important terms used, in order to minimize misunderstandings when referring to specific parts involved in the generation of data and the FIESTA-IoT concepts. The following definitions (listed in Table 2) were set regarding the domain area of FIESTA-IoT, and so are aligned with terminologies used in the Future Internet Research and Experimentation (FIRE) community and in reference to IoT-related projects (such as IoT-A).

Table 2 - Terminology and Definitions table

Term	Definition
Device	Technical physical component (hardware) with communication capabilities to other Information Technology (IT) systems. A device can be attached to, or embedded inside a physical entity, or monitor a physical entity in its vicinity [2]. The device could be: • **Sensor**: A sensor is a special device that perceives certain characteristics of the real world and transfers them into a digital representation [3]. • **Actuator**: An actuator is a mechanical device for moving or controlling a mechanism or system. It takes energy, usually transported by air, electric

	current, or liquid, and converts that into some kind of motion [3].
Discovery	Discovery is a service to find unknown resources/entities/services based on a rough specification of the desired result. It may be utilized by a human or another service. Credentials for authorization are considered when executing the discovery [2].
Domain	Refers to an application area where the meaning of data corresponds to the same semantic context. For instance, pressure in Water Management Domain may refer to water pressure on pipes while in Air Quality Domain it refers to atmospheric pressure.
Information	Content of communication; data and metadata describing data. The material basis is raw data, which is processed into relevant information, including source information (e.g., analogue and state information) and derived information (e.g., statistical and historical information) [4].
Measurement	The important data for the experimenter. It represents the minimum piece of information sent by a specific resource, which the experimenter needs in order to fulfil the objective of the experiment.
Metadata	The metadata is the additional information associated with the measurement, facilitating its understanding.
Physical Entity	Any physical object that is relevant from a user or application perspective [3]. Physical Entities are the objects from the real world that can be sensed and measured and they are virtualized in cyber-space using Virtual Entities.
Requirement	A quantitative statement of business-need that must be met by a particular architecture or work package [5].
Resource	Computational element that gives access to information about or actuation capabilities on a Physical Entity [3].
Testbed	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 [6].
Federated Testbeds	A Testbed federation or federated Testbeds is the interconnection of two or more independent Testbeds for the creation of a richer environment for experimentation and testing, and for the increased multilateral benefit of the users of the individual independent Testbeds [6].
Interoperability	The ability of two or more systems or components to exchange information and use the information that has been exchanged [7].
Experiment	Experiment is a test under controlled conditions that is made to demonstrate a known truth, examine the validity of a hypothesis, or determine the efficacy of something previously untried [8].
Semantic Interoperability	Semantic interoperability is the ability of computer systems to exchange data with unambiguous, shared meaning. Semantic interoperability is a requirement to enable machine computable logic, inference, knowledge discovery, and data federation between information systems.
Service	Services (Technology) are services designed to facilitate the use of technology by end users. These services provide specialized technology-oriented solutions by combining the processes/functions of software, hardware, networks, telecommunications and electronics.
Virtual Entity	Computational or data element representing a Physical Entity. Virtual Entities can be either Active or Passive Digital Entities [2].

1.5 Executive Summary

Testbeds generate plethora of data in their proprietary format. Differences in the data format results into interoperability issues between the Testbeds and much work has to be done in order to address this interoperability. The FIESTA-IoT platform provides a mechanism where interoperability between two or more Testbeds can be handled via a semantic model. The FIESTA-IoT Meta Cloud Architecture enables and enforces Testbeds to comply with such a specific semantic model.

This deliverable intends to give to diverse audience the right understanding about how the data produced by the Testbeds can be semantically annotated and stored, and finally be provided to the FIESTA-IoT platform so that the data can be used for experimentation. This makes this document one of the most essential documents to be studied and used by the Testbed owners. Nevertheless, in order to not forget other audiences of this document, it also provides Experimenters with the right understanding about what query experiments should make in order to get the data they are looking for. Note that, within FIESTA-IoT framework, experimenters will create experiments that would follow a specific Domain Specific Language (DSL). This language will incorporate the written query in order to retrieve the specific data from the FIESTA-IoT Meta Cloud.

In order to provide its audience the necessary aspects, this deliverable first presents the state of the art. This state of the art contains necessary background about related ontologies in the IoT domain, annotation tools and validation tools. After gathering such necessary knowledge, this document presents the methodology used to build the FIESTA-IoT Ontology. This is followed by an extensive description of the FIESTA-IoT ontology (which is the alignment of various existing IoT ontologies) along with the necessary definitions. This chapter not only provides an overview of the FIESTA-IoT ontology but also provides information on the M3-lite taxonomy that is used towards the definition of the related Quantity Kinds (QKs) and Units. Following the best practices and guidelines, the ontology is supported by an online documentation and some sample queries.

Moreover, this document provides to Testbed owners a reference annotation tool, developed by the existing In-house Testbeds that can be used as a reference by the Testbeds for the annotating their own IoT data. The FIESTA-IoT platform will validate such annotations utilizing dedicated validation tool based on FIESTA-IoT set policies.

Apart from this core part, this document also presents insights on composed data/service, mobility management and best practices to be followed by Testbed owners before producing annotated data.

As an added value, this document also provides some sample annotations for resources and measurements/observations from the In-house Testbeds. Further, the mapping between In-house Testbeds QKs and units to that of M3-lite taxonomy's QKs and units is also provided to facilitate the understanding of the semantic model to the external Testbeds owners who wish to join the federation.

2 RELATED WORK

In this section we present the related work pertaining to:

- Available IoT related ontologies that make potential candidate to be integrated in the final FIESTA-IoT Ontology,
- Potential ontology annotation tools that are currently available,
- Potential ontology validation tools that are currently available,
- Composition of Data/Service analysis.

2.1 Linked Open Vocabularies for Intern of Things (LOV4IoT)

To be able to build the FIESTA-IoT ontologies, a deep investigation of ontologies related to Internet of Things (IoT), Sensor Networks and Smart Cities has been done. The Linked Open Vocabularies for Internet of Things (LOV4IoT[1]), is an extension of the Linked Open Vocabularies (LOV[2]), a catalogue of ontologies. LOV4IoT is a dataset referencing almost three hundred ontology-based projects relevant for IoT. LOV4IoT classifies projects by domains. LOV4IoT classifies projects into nineteen domains (the number of ontologies present in the specified domain is marked in the brackets): smart home(46), smart energy(8), activity recognition(10), tourism(30), transportation(32), smart agriculture(17), weather forecasting(16), smart city(6), sensor networks(23), Internet of Things(20), healthcare(55), food/restaurant(30), affective science(6), music(6), environment(9), fire management(7), security(29), unit(5) and others(2). LOV4IoT highly guides us to demonstrate the need to unify existing ontologies rather than just designing a new one. In FIESTA-IoT context, LOV4IoT references three domains: IoT, sensor networks, and smart cities (see Figure 2, the three domains are marked with red box). In the smart city domain, no ontology follows semantic best practices[3] [9]. While in sensor networks domain, only W3C SSN ontology follows best practices those mentioned in [9]. Further, for the IoT domain, although SPITFIRE ontology follows best practices, this ontology is not online anymore.

Figure 2: Linked Open Vocabularies for Internet of Things (LOV4IoT)

[1] http://www.sensormeasurement.appspot.com/?p=ontologies
[2] http://lov.okfn.org/dataset/lov/
[3] http://sensormeasurement.appspot.com/?p=bestPractice

2.2 IoT-related Ontologies

This section analyses some of the ontologies designed for IoT and semantic sensor networks referenced within LOV4IoT that could be potentially reused within the FIESTA-IoT platform. Further, there are many surveys referencing ontologies relevant for IoT that are available and have their own vision [10], [11], [12]. However, we did neither find any recent survey describing the ontologies designed nor the ones specific to the IoT domain.

2.2.1 W3C SSN ontology

Semantic Sensor Networks (SSN) mainly focuses on interoperability and description of sensor networks to provide sensor discovery (see Figure 3). This forms a basis for our FIESTA-IoT ontology.

SSN is focused on the interoperability of the description of physical sensor networks to later ease the sensor discovery. “Semantics refers to the critical meaning of sensory data, sensor nodes and application requirements” and “effectively enabling the integration, exchange, and reuse of sensory data across various applications and multiple sensor nets” [13].

Analysis of the state of the art has been done regarding the semantic specification of sensors [11], [14]. There are many sensor ontologies available, such as:

- Coastal Environment Sensor Network (CESN) ontology [15].
- ontoSensor ontology, a prototype sensor knowledge repository [16], [17], [18].
- A service-oriented ontology for Wireless Sensor Networks (WSN) [19].
- An ontology for sensor networks describing the topology, network setting, sensor properties, dataflow and sensor network performance [20]
- Eid et al. [21], [22] and Avancha et al. ontologies [23].

However, these ontologies are mainly focused on the description of physical sensor networks such as sensor capabilities, location of the sensor (latitude and longitude), etc. They do not address the problem of describing the sensor data in an interoperable manner to ease the reasoning task. Compton et al. [12] reviewed all of these sensor ontologies[4] and compared them according to the following different criteria:

- Purpose of the ontology
- Status of the ontology: online, documentation, maintained, etc.
- Key concepts found in the ontology
- Adoption of the ontology
- Level of sophistication
- Weakest features

They came up with the W3C SSN ontology that is a synthesis of all the above-mentioned sensor ontologies [12]. SSN ontology defines high-level concepts for representing sensors, their observation and the surrounding environment.

However, the W3C SSN ontology has several limitations:

- It does not provide common terms to represent subclasses of `ssn:Sensor` type, measurement type, unit type or domain type (`ssn:FeatureOfInterest`). This shortcoming hinders machine automation.
- According to the ontology documentation[5], “the W3C SSN ontology does not describe domain concepts, time, locations, etc.” These concepts are intended to be included from other ontologies via Web Ontology Language (OWL) imports. This leads to

4 http://www.w3.org/2005/Incubator/ssn/wiki/Review_of_Sensor_and_Observations_Ontologies
5 http://www.w3.org/2005/Incubator/ssn/ssnx/ssn\#

interoperability issues between domain ontologies, since domain ontologies relevant for IoT are not standardized.

- In their final report, they explained that they “do not provide a basis for reasoning to ease the development of advanced applications”. This fact highlights the need to provide a common description of sensor measurements.
- Future works are to “standardize the SSN ontology to use it in a Linked Sensor Data context” and to “standardize the SSN ontology to bridge the IoT”.

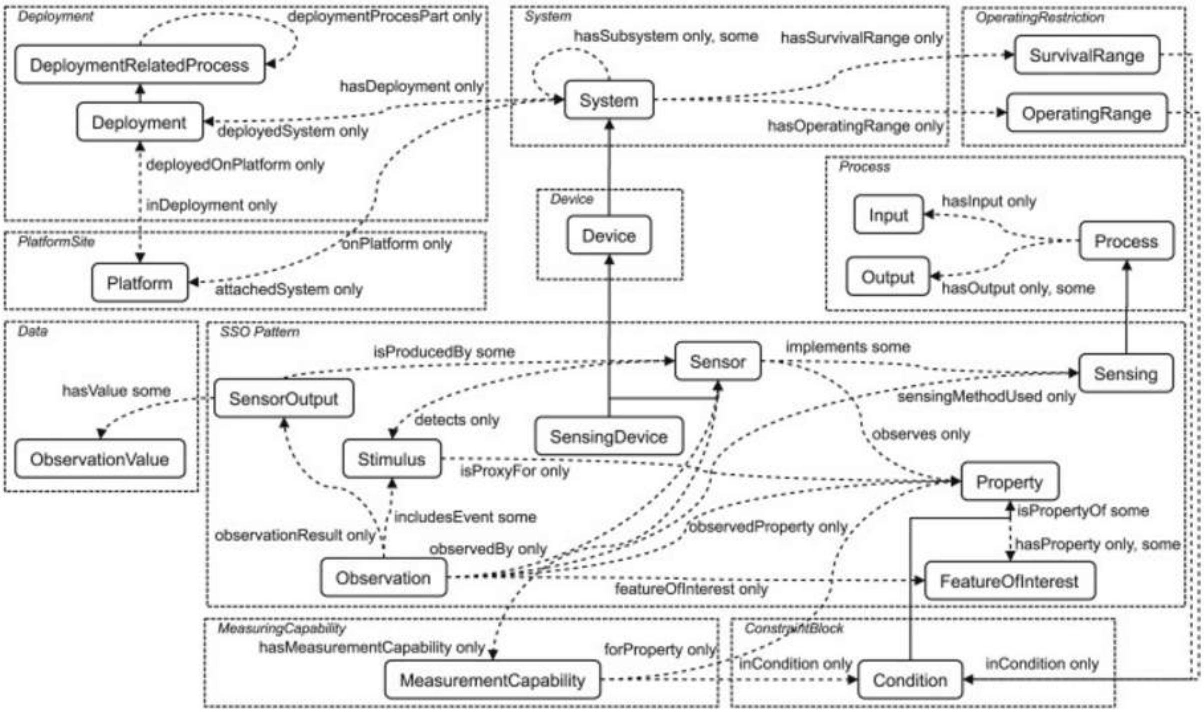


Figure 3: The W3C SSN ontology [12]

2.2.2 IoT-A Ontology

In addition to providing an architectural reference model for the IoT, IoT-A has also defined information models for its core concepts using ontologies (see Figure 4). The core concepts are the Resource, the Virtual Entity and the Service. It advances these concepts that were originally conceived in the SENSEI project [24], whereby the descriptions were not semantically oriented. The aim here was to address the enablement of linked open data to the descriptions by adopting semantic annotation. The overall model differs from SENSEI with the separation or decoupling of the Resource and the Service that exposes it to the Web. The idea here is that many Services can represent a Resource, each of which provides a different aspect about the information that is acquired from the Resource.

With this decoupling and following a strictly service-oriented approach, information relating to the phenomenon that is captured from the real/virtual world is kept in the Service. Here, the Resource becomes quite abstract. The only property that distinguishes a Resource is its “type”. A “type” can be a “Sensor”, “Actuator”, “Tag”, “Storage”, and “Network”. As in FIESTA-IoT we have opted for SSN based approach, we do not intend to use it for FIESTA-IoT ontology.

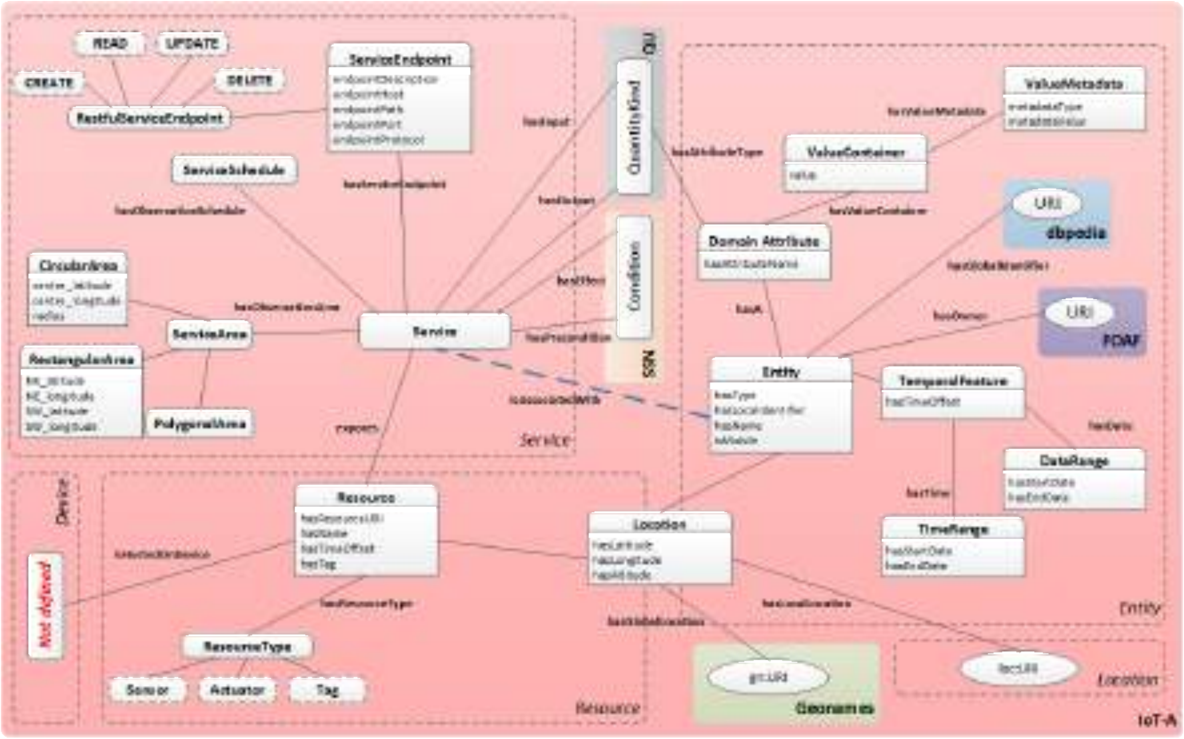


Figure 4: IoT-A Ontology

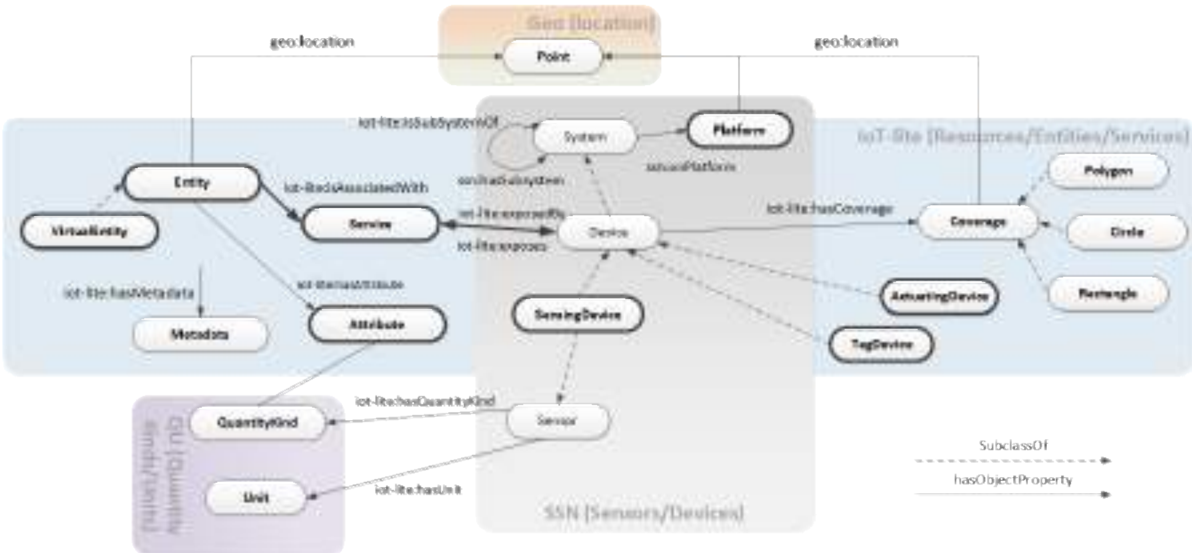


Figure 5: IoT-lite Ontology

2.2.3 IoT-lite ontology

The IoT-lite[6] (see Figure 5) is intended to be a “lite” ontology for describing devices, sensors, services and object (Virtual Entities). It is simpler than the IoT-A ontology but keeps its “spirit”. It also references a fragment of SSN (dealing with Sensor and Device descriptions). The IoT-lite ontology was developed as a part of EU FP7 FIWARE[7] project. However, in IoT-lite many updates have been performed to it so that it can be reused in FIESTA-IoT project.

[6] http://iot.ee.surrey.ac.uk/fiware/ontologies/iot-lite
[7] https://www.fiware.org

Thus, it forms one of the core ontology on which FIESTA-IoT ontology is built. Figure 5 shows the main concepts in the ontology and how they relate to each other.

2.2.4 Stream Annotation Ontology (SAO) ontology

The Stream Annotation Ontology[8] (SAO) (see Figure 6) is an extension of the SSN ontology and the PROVO ontology[9] to allow the representation of real-time data streams, analysed data streams and stream events [25]. It links to additional ontologies like the Quality Of Information ontology (QoI[10]) and the Timeline[11] ontology. The core concept of `StreamData` extends the `ssn:Observation` class, and can hold data with respect to a "point" in time or an array of data over a "segment" in time. We intend to use SAO ontology and extend FIESTA-IoT ontology for the next release of FIESTA-IoT ontology.

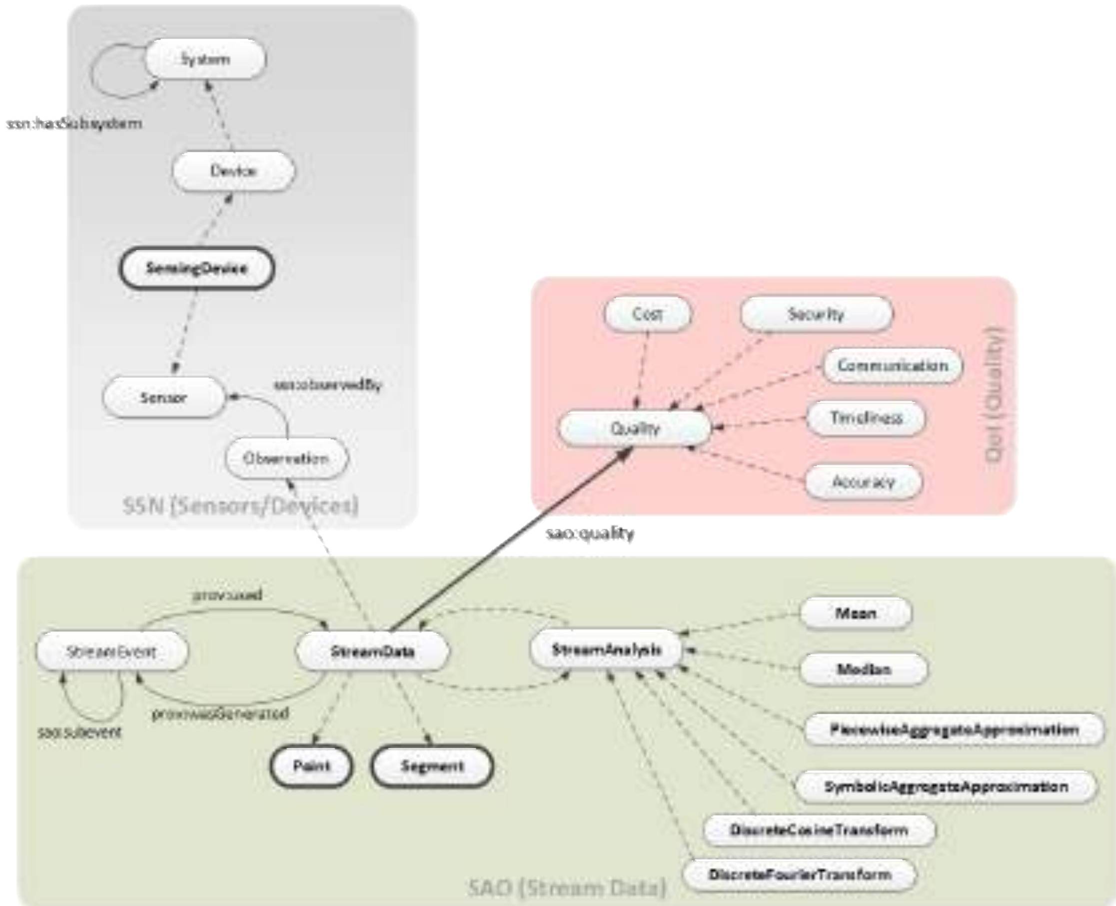


Figure 6: SAO Ontology [25]

2.2.5 OpenIoT Ontologies

Sensors – or sensing devices in general – are the most relevant instances of Internet-connected objects in the focus of OpenIoT ontology. As a consequence, the OpenIoT ontology is adopting the W3C SSN ontology [12] as its core component (such as `Sensor`, `ObservationValue` and `FeatureOfInterest`), and extends it by adding missing concepts required for the OpenIoT-specific requirements. In Figure 7, it can be seen a simplified OpenIoT data model that includes the interaction with the above concepts.

[8] http://iot.ee.surrey.ac.uk/citypulse/ontologies/sao/sao
[9] https://www.w3.org/TR/prov-o/
[10] http://purl.oclc.org/NET/UASO/qoi
[11] http://motools.sf.net/timeline/timeline.html

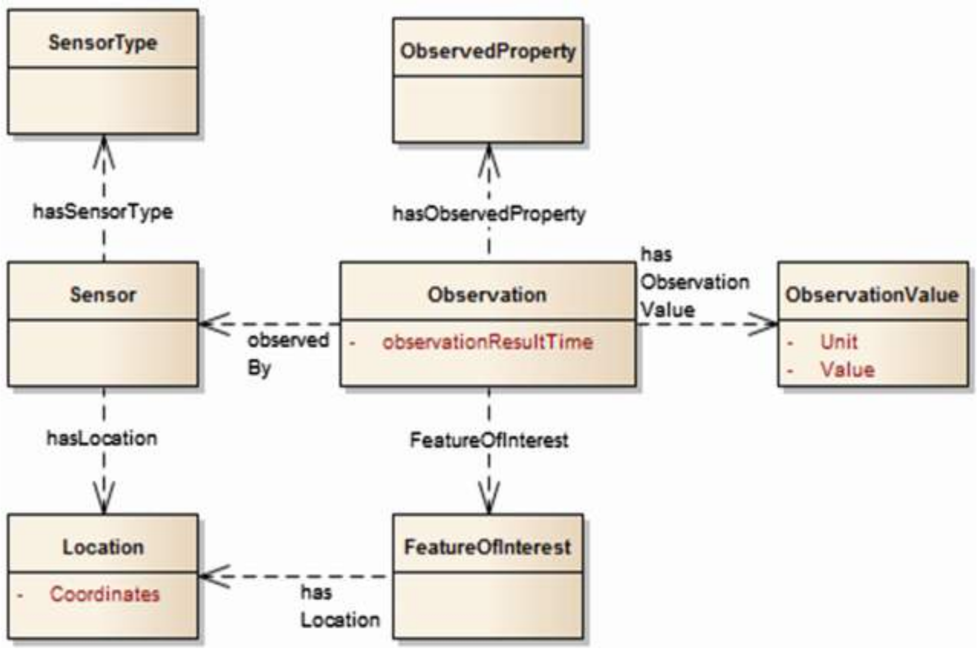


Figure 7: Simplified OpenIoT data model [26]

2.2.5.1 OpenIoT Vocabulary

OpenIoT is making use of existing vocabularies that are mentioned in the "Ontology Namespace" table in the starting of this deliverable. The reused vocabularies for in OpenIoT are [27]:

```
@prefix ssn:< http://purl.oclc.org/NET/ssnx/ssn#> .
@prefix spt:<http://spitfire-project.eu/ontology/ns/> .
@prefix dul:<http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:<http://www.w3.org/2002/07/owl#> .
@prefix xsd:<http://www.w3.org/2001/XMLSchema#> .
@prefix ex:<http://www.example.org/ns#> .
@prefix openiot:<http://www.openiot.eu/ontology/ns#> .
```

Additionally to existing vocabularies that have been reused, OpenIoT ontology introduces some new concepts that haven't been covered by the existing vocabularies and may be of interest for the FIESTA-IoT ontology. Such concepts can be explored towards adding them in FIESTA-IoT ontology. These concepts include the notion of Virtual sensors and Utility metrics as described below.

2.2.5.2 Virtual sensor

Virtual sensor [27] – the basic concept in Global Sensor Network (GSN) that is one core element of the OpenIoT platform – represents new data sources created from live data. These virtual sensors can filter, aggregate or transform the data. From an end-user perspective, both virtual and physical sensors are very closely related concepts since they both, simply speaking, measure data. Therefore, the concept of a virtual sensor is as a subclass of the sensor concept as defined in the SSN ontology (see below).

```
openiot:VirtualSensor rdfs:subClassOf ssn:Sensor ;
rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
```

2.2.5.3 Utility Metrics

In OpenIoT, utility metrics are used in a variety of utility-based algorithms for resource management, utility-driven privacy and utility-driven security mechanisms. In addition, utility metrics serve as a basis for accounting and management of Service Level Agreements (SLA) between the OpenIoT services and end users. Three categories of utility metrics are identified which are described below:

2.2.5.3.1 Utility Metrics for Physical Sensors

The following metrics [27] have been formulated for physical sensors. They essentially cover the typical metrics derived from a technical perspective, such as quality, energy consumption, bandwidth, data volume and trustworthiness. The above mentioned properties are defined via `ssn:Accuracy`, `ssn:Sensitivity`, `spt:Energy`, `openiot:Bandwidth`, `openiot:DataVolume` and `openiot:Trustworthiness`. More description about the metrics is available in the Appendix I – Openiot Utility Metrics.

2.2.5.3.2 Utility Metrics for Virtual Sensors

The following list [27] provides utility metrics that are applicable to virtual sensors. Hence, OpenIoT defines a vocabulary for the set of utility metrics that can be used for virtual sensors. The vocabulary consists of data volume, bandwidth, time of the usage session, virtual sensor location, virtual sensor task, number of physical sensors, type of physical sensor, and user defined costs. The above mentioned properties are defined via `ssn:Accuracy`, `ssn:Sensitivity`, `openiot:SessionTime`, `openiot:VirtualSensorLocation`, `openiot:VirtualSensorTask`, `openiot:subSensorsCount`, `ssn:Sensor`, and `openiot:SensorUtility`. More description about the metrics is available in the Appendix I – Openiot Utility Metrics.

2.2.5.3.3 Utility Metrics for a Sensor Network and Application Service Level

The utility of sensor network and the application service level can be measured based on the various parameters such as system lifetime, latency, quality, delay, delay variation, bandwidth, capacity, throughput, hot count, ease of deployment, reliability, survivability, scalability, resource optimization, cost efficiency, relevance and confidentiality. The above mentioned properties are defined via `ssn:SystemLifetime`, `ssn:Latency`, `spt:NetworkQuality`, `openiot:Delay`, `openiot:DelayRange`, `openiot:NetworkBandwidth`, `openiot:NetworkCapacity`, `openiot:NetworkThroughput`, `openiot:LinkCost`, `openiot:EaseOfDeployment`, `openiot:Reliability`, `openiot:Survivability`, `openiot:Scalability`, `openiot:ResourceOptimization`, `openiot:CostEfficiency`, `openiot:Relevance`, and `openiot:Confidentiality`. More description about the metrics is available in the Appendix I – Openiot Utility Metrics.

2.2.5.4 OpenIoT ontology graph representation

Figure 8 below shows the OpenIoT ontology as a graph on a high level, i.e. not all defined subclasses. All subclasses for `MeasurementProperty`, `NetworkProperty` and `OperatingProperty` are shown in Figure 9 - Figure 11 below.

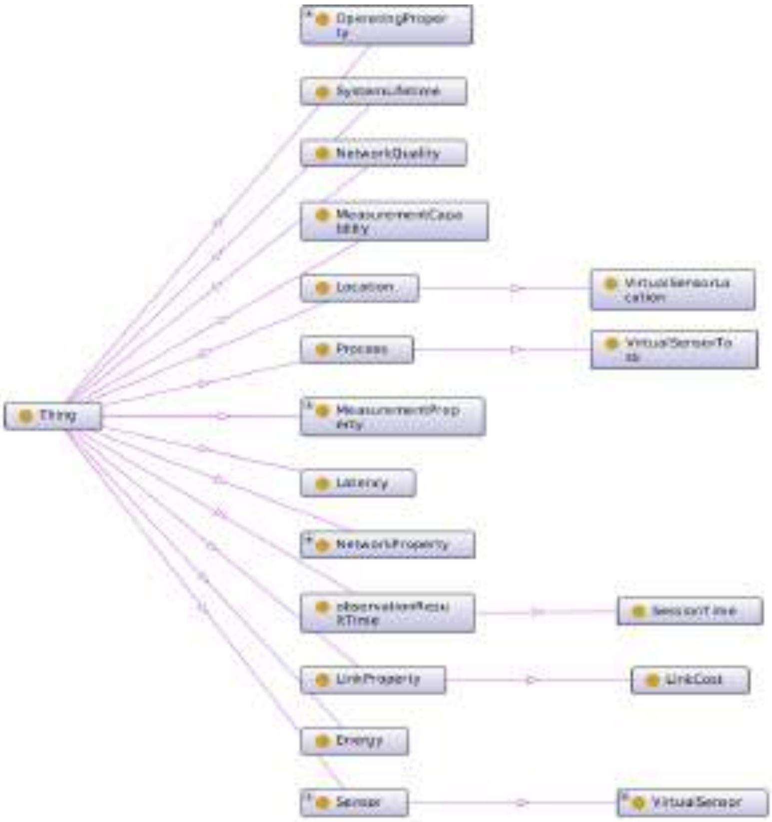


Figure 8: High-level graph representation of OpenIoT ontology [27]

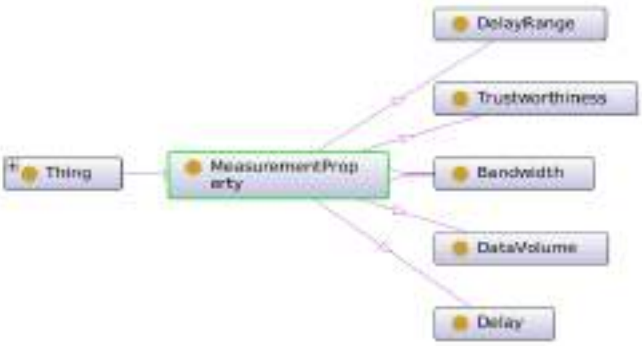


Figure 9: Graph representation of `MeasurementProperty` [27]

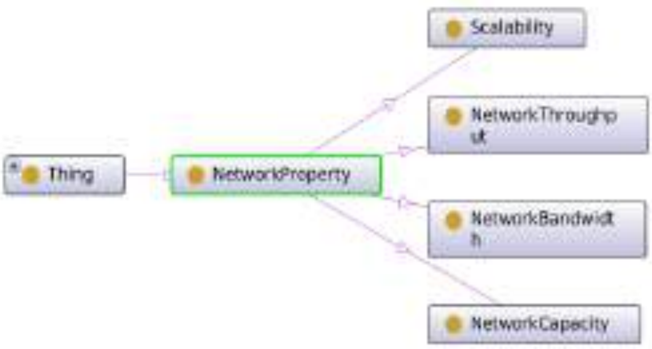


Figure 10: Graph representation of `NetworkProperty` [27]

Figure 11: Graph representation of `OperatingProperty` [27]

2.2.6 Machine-to-Machine Measurement (M3) Nomenclature and Ontology

Since sensor data is coming from heterogeneous Testbeds that use different vocabularies and annotations, they are not interoperable a priori. For this reason, the **Machine-to-Machine Measurement (M3)** nomenclature has been designed to make the data interoperable and explicitly add the context when required to avoid any ambiguities [28]. For instance, a temperature could be a body temperature or the environment temperature. To achieve this step, semantic web technologies have been used, and the M3 ontology has been designed to implement the M3 nomenclature for the following reasons:

- to ease interoperability,
- to add explicit sensor metadata descriptions,
- to employ reasoning engine to infer new knowledge,
- to reuse domain knowledge, and
- to provide a flexible and easy way to update the M3 nomenclature.

The M3 nomenclature has been implemented as an ontology to provide a basis for reasoning and interlinking domains, one with each other. The M3 ontology synthesizes and unifies all terms to describe sensors, measurements, actuators and domains found in existing IoT projects. For instance, precipitation and rainfall sensors represent the same type of sensor. The uniform descriptions above-mentioned are fundamentally necessary to develop cross-domain interoperable semantic-based applications and services. A common nomenclature is described in the following subsections. Note that the list provided in this document is not exhaustive. Furthermore, for FIESTA-IoT ontology we have created a “lite” version of the M3 ontology (see Section 3.3.4)

2.2.6.1 M3 an extension of W3C SSN

The M3 ontology is an extension of the W3C SSN ontology. More precisely, it extends `ssn:ObservationValue`, `ssn:FeatureOfInterest` and `ssn:Sensor` concepts. M3 deal not only with sensors but also with Radio Frequency Identification (RFID) tags, actuators, etc. M3 has been initially designed to explicitly describe and unify heterogeneous measurements. For instance, a same measurement can be described in a different way; a measurement has a name (e.g., temperature or temp), a value (e.g., 39) and a unit (e.g., DegreeCelsius or Celsius). To ensure interoperability among sensor data, the reuse of the M3 ontology is preconized. The uniform M3 descriptions of sensor, measurements and domains have already been communicated to oneM2M WG-5 (MAS)[12] [29].

[12] http://onem2m.org/MAShome.cfm

2.2.6.2 Unifying Sensor measurement and unit names

Table 3 presents such a common nomenclature for the sensors used in weather domain. The first column of the Table 3 is the recommended uniform sensor and measurement name, various other names are listed in the second column and units in the third column.

The M3 ontology has been designed by analysing the Linked Open Vocabularies for IoT (LOV4IoT) dataset[13] (see Section 2.1). Most of the sematic-based projects in LOV4IoT provide an ontology describing sensors, measurement names and units employed in IoT domain.

Table 3: M3 uniform description for sensors, observations and units in the weather domain

M3 sensor/ measurement name	Description, other names (synonyms)	M3 Unit
HumiditySensor/Humidity	Hygrometer, humidity sensor, moisture sensor, soil moisture probes	Percent
WindDirectionSensor/ WindDirection	Wind direction	DegreeAngle
SunPositionDirectionSensor/ SunPosition	Sun position direction to detect east, west, south, north	DegreeAngle
AtmosphericPressureSensor/ AtmosphericPressure	Atmospheric pressure sensor, Barometer, barometric pressure sensor	Pascal
CloudCoverSensor/ CloudCover	Cloud cover sensor	Okta
SunPositionElevationSensor/ SunElevation	Sun position elevation to detect (twilight, day, night, etc.)	DegreeAngle
SolarRadiationSensor/ SolarRadiation	Solar radiation sensor, par (photo synthetically active radiation) sensor, sun light, solar sensors, sun's radiation intensity	WattPerMeterSquare
VisibilitySensor/Visibility	Visibility sensor to detect fog	Miles, Meter
Thermometer, AirThermometer/Temperature	Thermometer, temperature sensor, thermistor	DegreeCelsius
LightSensor/Luminosity	Light, luminosity, illuminance, lighting	Lux
PrecipitationSensor/ Precipitation	Precipitation sensor, rainfall sensor, rain fall, pluviometer, rain, rainfall gauge	MilimeterPerHour
WindSpeedSensor/ WindSpeed	Wind speed sensor, wind velocity sensor, anemometer	MeterPerSecond

The entire M3 nomenclature is available online[14]. The classification of device type has been performed in the IoT domains as well as in health care, smart home, transportation, agriculture, air quality measuring and with actuators.

13 http://sensormeasurement.appspot.com/?p=ontologies
14 http://www.sensormeasurement.appspot.com/documentation/NomenclatureSensorData.pdf

2.2.6.3 Unifying IoT applicative domains

The M3 ontology classifies observations, sensors, units in 12 IoT domains, namely: building automation, health, weather, agriculture, environment, emotion, transport, energy, tourism, location, city, and tracking good. The M3 ontology extends the `ssn:FeatureOfInterest` class to do such classification. Table 4 shows uniform IoT applicative domain names.

Table 4: M3 uniform description for IoT applicative domain names

M3 Domain name	Description, other names (synonyms)
BuildingAutomation (subclass: Activity)	Smart home, building automation, or building or room (kitchen, bathroom, living room, dining room)
Health	Healthcare
Weather	Weather forecasting, meteorology
Agriculture	Agriculture, smart farm, garden
Environment (subclass: Fire)	Environment (earthquake, flooding, forest fire, air pollution)
Emotion	Affective science, emotion, mood, emotional state; brain wave
Transport	Intelligent transportation systems (ITS), smart car/vehicle, transportation
EnergyFOI	Smart grid, smart energy
Tourism	Tourism
Location	Location, place, GPS coordinates
City	Smart city, city automation, public lighting
TrackingGood (sub-classes: TrackingFood, TrackingCD)	Tracking RIFD goods
Generic	Others

2.2.6.4 M3 aligned with existing ontologies

The M3 ontology is a first step towards building an "ontology network" of IoT ontologies. This precursor idea aims to navigate from an ontology to another since they the knowledge designed within ontologies complement each other. For this reason, M3 has been interconnected with numerous ontologies to design an "IoT ontology network". As an example, M3 is interconnected with the following ontologies, but the list is not exhaustive, since M3 is constantly updated and interconnected with new ontologies:

- IoT ontologies (OpenIoT, SPITFIRE[15])
- Semantic Sensor Networks ontologies (W3C SSN, ontoSensor)
- Unit ontologies (QUDT, QUDT_unit, Measurement Unit Ontology (muo))
- Location ontologies (WGS84)
- Domain ontologies
    - Smart home (Smart Home Weather ontology (shw), etc.)
    - Weather (Ontology for Meteorological sensors (aws), Smart Home Weather ontology, etc.)
    - etc.

Ontology namespaces have been referenced in the ontology namespace table described above in this deliverable.

[15] http://spitfire-project.eu/ontology/ns

Each time that a new ontology is referenced within LOV4IoT, M3 analyses it and when common concepts, instances or properties are found matching with M3 descriptions of sensors, measurements, units or domains, a link is explicitly added (`rdfs:subClassOf`, `owl:sameAs`, etc.) within the M3 ontology.

2.2.6.5 M3 validated with semantic web tools

The M3 ontology has been validated with the SSN validator and other semantic web tools such as Oops[16], TripleChecker[17], Vapour[18], LOV, etc. Based on this experience, we planned to follow best practices as much as possible to decide which ontologies can be reused within FIESTA and also to ensure that the FIESTA ontology follows best practices to ease interoperability. More information regarding the set of best practices have been disseminated within W3C Web of Things [30], World Forum for Internet of Things [9], and the standardization OneM2M Management, Abstraction and Semantics (MAS) Working Group [29]. A summary of tools with all relevant tools is also available[3].

2.2.7 VITAL ontology

VITAL[19] is an EU project that is deployed in two cities: Istanbul and London [31]. The VITAL project aims to provide semantic interoperability among IoT applications and projects to reduce the costs associated while developing new smart city applications. VITAL reuses the work achieved by OpenIoT and the W3C SSN ontology. However, VITAL is not aligned to existing IoT ontologies such as the ones presented in Section 2.2 and developed in the past by the FIESTA-IoT partners (e.g., IoT-lite, SAO, M3, etc.).

VITAL uses DCN, DUL, WGS84, hREST, MSM, Web Ontology Language, RDF Schema ontology, Semantic Annotations for WSDL and XML Schema ontology, SSN, Time, WSL, XSD, QUDT, FOAF, S4AC, OTN ontologies (see “Ontology Namespace” Table). Reusing ontologies ensure better interoperability and reduce development costs.

2.2.8 Hachem et al. Ontology

Hachem et al. [32] proposed three ontologies: the “Device ontology” to represent physical things such as hardware devices, the “Domain ontology” to model the physical concepts and relationships such as those provided by formulas in the domain of physics and mathematics and conversion for units, and the “Estimation ontology”. The main focus of [32] was to quantify features of sensing and actuation tasks and to help better understand those tasks. Hachem et al. [32] reused the W3C SSN and SWEET[20] ontology. Moreover, this ontology is not maintained. However, for FIESTA-IoT that forms a meta cloud on top the Testbeds and queries the data available with Testbeds, such ontology might not be needed. In case of the requirements coming up from the Testbeds joining in future, the ontology can be explored to find the potential use and integration within the context of FIESTA-IoT.

2.2.9 SmartSantander Taxonomy with QuantityKinds (phenomena) and Units (unit of measurement)

During the first implementation phase of the SmartSantander[21] Testbed, the annotation of the resources’ capabilities (i.e. quantity kinds and units) was made according to a proprietary

[16] http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/technologies/292-oops
[17] http://graphite.ecs.soton.ac.uk/checker/
[18] http://validator.linkeddata.org/
[19] http://vital-iot.eu
[20] https://sweet.jpl.nasa.gov
[21] http://smartsantander.eu/

and non-semantic format. However, the growing impact of semantic oriented projects (among which FIESTA-IoT is expected to stand out as a spearhead solution) and the emergence of potential standardized products (e.g. oneM2M) has tilted the balance towards having a semantic approach. Thus, after a thorough analysis stage, most of the physical phenomena and their matching units of measurement were mapped according to a number of different and well-known reputed ontologies, such as SWEET[20], QU-Rec20[22], and QUDT[23], etc. As a tangible result of this work, Table 5 and Table 6 gather all the quantitykinds (QKs) and units of measurement that are present throughout the different types of sensors and resources deployed over the SmartSantander platform, linked to the ontology Internationalized Resource Identifier (IRI) and its corresponding definition. In Table 5 and Table 6 we list classes and instances (of the classes) that reflect QKs.

Table 5: SmartSantander QuantityKinds and well-known ontologies connection

QuantityKind	Ontology Ref	Definition
acceleration:instantaneous	http://purl.oclc.org/NET/ssnx/qu/dim#Acceleration	The rate of change with time of the velocity vector of a particle.
activePower	-	The product of the voltage across a branch of an alternating-current circuit and the component of the electric current that is in phase with the voltage.
atmosphericPressure	http://purl.oclc.org/NET/ssnx/qu/quantity#atmosphericPressure	The pressure exerted by the atmosphere as a consequence of gravitational attraction exerted upon the column of air lying directly above the point in question.
batteryLevel	http://www.w3.org/2007/uwa/context/deliveryContext.owl#BatteryLevel	This property holds the percentage of the battery capacity that remains available.
chemicalAgentAtmosphericConcentration:03	-	The concentration of ozone (O_3) gas suspended in an atmosphere.
chemicalAgentAtmosphericConcentration:airParticles	-	The concentration of air particles suspended in the atmosphere.
chemicalAgentAtmosphericConcentration:CO	-	The concentration of CO gas suspended in an atmosphere.
chemicalAgentAtmosphericConcentration:NO2	-	The concentration of NO_2 gas suspended in an atmosphere.
direction:heading	-	The course or direction in which an object (vehicle, person, ...) is pointing or moving.
electricCurrent	http://qudt.org/vocab/quantity#ElectricCurrent	Electric current is the flow of electric charge. It is a base quantity in the International System of Units. Electric current is electric charge divided by time. Electric Current is the flow (movement) of electric charge. The amount of electric current through some surface, e.g., a section through a copper conductor, is defined as the amount of electric charge flowing through that surface over time.
electricField:1800Mhz	http://qudt.org/vocab/quantity#ElectricField	A field of force associated with a moving electric charge equivalent to an electric field and a magnetic field at right angles

[22] https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu-rec20.html
[23] http://qudt.org/

		to each other and to the direction of propagation. Applied to the 1800MHz UHF band.
electricField:2100Mhz	http://qudt.org/vocab/quantity#ElectricField	A field of force associated with a moving electric charge equivalent to an electric field and a magnetic field at right angles to each other and to the direction of propagation. Applied to the 2100MHz UHF band.
electricField:2400Mhz	http://qudt.org/vocab/quantity#ElectricField	A field of force associated with a moving electric charge equivalent to an electric field and a magnetic field at right angles to each other and to the direction of propagation. Applied to the 2400MHz UHF band.
electricField:900Mhz	http://qudt.org/vocab/quantity#ElectricField	A field of force associated with a moving electric charge equivalent to an electric field and a magnetic field at right angles to each other and to the direction of propagation. Applied to the 900MHz UHF band.
electricPotential	http://purl.oclc.org/NET/ssnx/qu/dim#ElectricPotential	Electric potential is the potential energy per unit charge associated with static (time-invariant) electric field.
fillLevel:gasTank:1	-	Ratio between volume of combustible exhaust fluid and the total volume of diesel exhaust fluid storage container. This is the value for the first tank of the vehicle.
fillLevel:gasTank:2	-	Ratio between volume of combustible exhaust fluid and the total volume of diesel exhaust fluid storage container. This is the value for the second tank of the vehicle.
fillLevel:wasteContainer	-	Ratio between the current fill level and the total capacity of a waste container.
fuelComsumption:total	-	Accumulated amount of fuel used during vehicle operation.
fuelConsumption:instantaneous	-	A measure that displays the instantaneous fuel consumption of a vehicle during its operation.
illuminance	http://purl.oclc.org/NET/ssnx/qu/quantity#illuminance	Illuminance is the total luminous flux incident on a surface, per unit area. It is a measure of the intensity of the incident light, wavelength-weighted by the luminosity function to correlate with human brightness perception.
mass	http://purl.oclc.org/NET/ssnx/qu/quantity#mass	Mass is a property that is a constant that indicates the resistance of a material against acceleration.
mileage:distanceToService	http://purl.org/uco/ns#mileage	The distance that can be travelled by the vehicle before the next service inspection is required. A negative distance is transmitted if the service inspection has been passed.
mileage:total	http://purl.org/vso/ns#mileageFromOdometer	The total distance travelled by the particular vehicle since its initial production.
motionState:vehicle	-	Indicates whether motion of the vehicle

		is detected or not.
position:altitude	http://sweet.jpl.nasa.gov/2.3/propSpaceHeight.owl#Altitude	The vertical distance above mean sea level.
position:latitude	http://sweet.jpl.nasa.gov/2.3/reprSpaceCoordinate.owl#Latitude	The angular distance north or south from the equator of a point on the earth's surface, measured on the meridian of the point.
position:longitude	http://sweet.jpl.nasa.gov/2.3/reprSpaceCoordinate.owl#Longitude	The angular distance east or west on the earth's surface, measured by the angle contained between the meridian of a particular place and some prime meridian, as that of Greenwich, England, and expressed either in degrees or by some corresponding difference in time.
presenceState:driverCard:1	-	Indicates the presence of the first driver card.
presenceState:driverCard:2	-	Indicates the presence of the second driver card.
presenceState:parking	-	The presence or absence of a vehicle parked.
rainfall	http://sweet.jpl.nasa.gov/ontology/property.owl#Rainfall	The depth of precipitation (water-equivalent) that accumulated over a measurement time quantity.
reactivePower	http://purl.oclc.org/NET/ssnx/qu/dim#ReactivePower	The portion of electricity that establishes and sustains the electric and magnetic fields of alternating-current equipment.
relativeHumidity	http://sweet.jpl.nasa.gov/ontology/property.owl#RelativeHumidity	The ratio of vapours pressure to saturation vapour pressure, where vapour pressure is the pressure exerted by the molecules of water vapour and saturation vapour pressure is the pressure exerted by molecules of water vapour in AIR that has attained saturation.
roadOccupancy	-	Ratio of time on which a road lane section is occupied by vehicles within a given period of time.
rotationalSpeed:engine	-	Rotational speed is a property that is the rate of rotation of a material around an axis, in this case the engine cylinders.
soilMoistureTension	-	The force per unit area required to remove film water from soil.
solarRadiation:par	-	Photosynthetically Active Radiation (PAR) is the amount of light available for photosynthesis, which is light in the 400 to 700 nanometre wavelength range. PAR changes seasonally and varies depending on the latitude and time of day.
soundPressureLevel:ambient	decibelA	Sound pressure level is a logarithmic measure of the RMS sound pressure of a sound relative to a reference value, the threshold of hearing. The reference sound pressure was chosen conventionally to correspond to the quietest sound at 1000Hz that the

		human ear can detect (20uPa). In this case, the specific parameter is measured in an open environment.
speed:average	http://purl.oclc.org/NET/ssnx/qu/quantity#speed	A measure of the average rate of motion of an object.
speed:instantaneous	http://purl.oclc.org/NET/ssnx/qu/quantity#speed	A measure of the instantaneous rate of motion of an object.
speed:median	http://purl.oclc.org/NET/ssnx/qu/quantity#speed	A measure of the median rate of motion of an object.
temperature:ambient	http://purl.oclc.org/NET/ssnx/qu/quantity#temperature	The temperature of the air that would be indicated by a thermometer exposed to the air at a location sheltered from direct solar radiation
temperature:engine	http://purl.oclc.org/NET/ssnx/qu/quantity#temperature	The temperature of a vehicle engine.
temperature:soil	http://purl.oclc.org/NET/ssnx/qu/quantity#temperature	Soil temperature is the bulk temperature of the soil, not the surface (skin) temperature.
temperature:wasteContainer	http://purl.oclc.org/NET/ssnx/qu/quantity#temperature	The temperature of the air that would be indicated by a thermometer exposed to the air inside a waste container.
timeRelatedState:driver:1	-	Indicates if the first driver approaches or exceeds working time limits (or other limits).
timeRelatedState:driver:2	-	Indicates if the second driver approaches or exceeds working time limits (or other limits).
timestamp	http://sweet.jpl.nasa.gov/2.3/propTime.owl#Timestamp	Sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day, sometimes accurate to a small fraction of a second. This representation should be encoded following ISO8601.
trafficCongestion	-	Descriptive level of road traffic congestion within a defined area
trafficIntensity	-	The intensity of a traffic flow is the number of vehicles passing a cross section of a road in a unit of time.
vehicleOverspeedState	-	Indicates whether the vehicle is exceeding the legal speed limit.
windDirection	http://purl.oclc.org/NET/ssnx/meteo/WM30#WindDirection	The geodetic azimuth of the direction from which the wind is blowing.
windSpeed	http://sweet.jpl.nasa.gov/2.3/propSpeed.owl#WindSpeed	The ratio of the distance covered by moving air to the time quantity taken to cover it.
workingState:driver:1	-	State of work of the first driver as defined in the FMS standard.
workingState:driver:2	-	State of work of the second driver as defined in the FMS standard.

Table 6: SmartSantander (Base) Units and well-known ontologies connection

Unit	Ontology Ref	Definition
ampere	http://purl.oclc.org/NET/ssnx/qu/unit#ampere	The ampere is the International System of Units (SI unit) for measuring an electric current which is the flow of electric charges through a surface at the rate of one coulomb per second.
bar	http://qudt.org/vocab/unit#Bar	Metric unit of atmospheric pressure equal to 14.50 pounds per square inch (lb/in^2), 1.02 kilograms per square centimetre (kg/cm^2), 29.53 inches of mercury (in Hg), or 0.9869 atmosphere.
cubicMeterPerSecond	-	The SI unit of volumetric flow rate equal to that of a steer or cube with sides of one metre in length exchanged or moving each second.
decibel	http://purl.oclc.org/NET/ssnx/qu/unit#decibel	A ratio unit which is an indicator of sound power per unit area.
decibelA	-	A relative loudness of sounds in air unit as perceived by the human ear.
degreeAngle	http://qudt.org/vocab/unit#DegreeAngle	A plane angle unit which is equal to 1/360 of a full rotation or π/180 rad.
degreeCelsius	http://purl.oclc.org/NET/ssnx/qu/unit#degreeCelsius	A temperature unit that is equal to 1kelvin degree. However, they have their zeros at different points. The centigrade scale has its zero at 273.15K.
dimensionless	http://purl.oclc.org/NET/ssnx/qu/dim#Dimensionless	Without dimensions; having no appreciable or noteworthy extent. Without physical meaning.
index	-	Arbitrary indirect reference that should be translated into meaningful measurements by using the corresponding decoding algorithm detailed in the resource description. In this case the returned values can only take certain values from a finite set.
kilogram	http://purl.oclc.org/NET/ssnx/qu/unit#kilogram	The SI unit of mass, it is equal to the mass of the international prototype of the kilogram.
kilometrePerHour	http://purl.oclc.org/NET/ssnx/qu/unit#kilometrePerHour	Kilometre per hour is a unit of speed defined as kilometre divided by hour.
kilowattHour	http://purl.oclc.org/NET/ssnx/qu/unit#kilowattHour	A measure of electrical energy equivalent to a power consumption of one thousand watts for one hour.
litre	http://purl.oclc.org/NET/ssnx/qu/unit#litre	A metric unit of capacity defined as the volume of one kilogram of water under standard conditions. It is equal to 1,000cm^3 .
litrePer100Kilometres	-	A consumption unit which is equal to the one of a vehicle which needs 1 fuel litre in order to traverse 100kilometres.
lumen	http://purl.oclc.org/NET/ssnx/qu/unit#lumen	The SI unit of luminous flux, equal to the amount of light emitted per second in a unit solid angle of one steradian from a uniform source of one candela.
lux	http://purl.oclc.org/NET/ssnx/qu/unit#lux	The SI unit of illuminance, equal to one lumen per square metre.
metre	http://purl.oclc.org/NET/ssnx/qu/unit#metre	The SI unit of distance, the metre is the length of the path travelled by light in vacuum during a time interval of 1/299792458 of a second.
metrePerSecond	http://purl.oclc.org/NET/ssnx/qu/unit#metrePerSecond	The metre per second squared is the unit of acceleration in the SI units. As a derived unit it is composed from the SI base units of length,

		the metre, and time, the second. As acceleration, the unit is interpreted physically as change in velocity or speed per time interval, i.e. metre per second per second and is treated as a vector quantity.
metrePerSecondSquared	http://purl.oclc.org/NET/ssnx/qu/unit#metrePerSecondSquared	The metre per second squared is the unit of acceleration in the SI units. As a derived unit it is composed from the SI base units of length, the metre, and time, the second. As acceleration, the unit is interpreted physically as change in velocity or speed per time interval, i.e. metre per second per second and is treated as a vector quantity.
microgramPerCubicMetre	http://purl.oclc.org/NET/ssnx/qu/unit#microgramPerCubicMetre	Microgram per cubic metre is a unit of density defined as microgram divided by cubic metre. Milligram per cubic metre is a derived unit in the International System of Units.
milligramPerSquareMetre	http://purl.oclc.org/NET/ssnx/qu/unit#milligramPerSquareMetre	A dose unit which is equal to 1 milligram of a toxic or pharmaceutical substance per square meter of surface area of the recipient subject
millivoltPerMetre	http://purl.oclc.org/NET/ssnx/qu/unit#millivoltPerMetre	An electromagnetic field of 1 V/m is represented by a potential difference of 1 V existing between two points that are 1 m apart.
pascal	http://purl.oclc.org/NET/ssnx/qu/unit#pascal	The Pascal is the unit of pressure or stress in the International System of Units (SI), equivalent to one newton (1N) of force applied over an area of one meter squared ($1m^2$). That is, $1Pa = 1Nm^2$. Reduced to base units in SI, one Pascal is one kilogram per meter per second squared; that is, $1Pa = 1kgm^{-1}s^{-2}$.
percent	http://purl.oclc.org/NET/ssnx/qu/unit#percent	A dimensionless ratio unit which denotes numbers as fractions of 100.
radianPerSecond	http://purl.oclc.org/NET/ssnx/qu/unit#radianPerSecond	The radian per second is defined as the change in the orientation of an object, in radians, every second. The radian per second is the SI unit of angular (rotational) speed.
revolutionPerMinute	-	Revolutions per minute (abbreviated rpm, RPM, rev/min, r/min) is a measure of the frequency of rotation, specifically the number of rotations around a fixed axis in one minute. It is used as a measure of rotational speed of a mechanical component.
scale	-	Arbitrary indirect reference that should be translated into meaningful measurements by using the corresponding decoding algorithm detailed in the resource description. In this case the returned values are part of a continuous variable which can take any numeric value.
second	http://purl.oclc.org/NET/ssnx/qu/unit#secondUnitOfTime	The second is the duration of 9.192.631.770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium 133 atom.
var	http://purl.oclc.org/NET/ssnx/qu/unit#var	In electric power transmission and distribution, volt-ampere reactive (var) is a unit in which reactive power is expressed in an AC electric power system.
vehiclePerMinute	-	Number of vehicles that traverse a concrete region of the space in one minute.

volt	http://purl.oclc.org/NET/ssnx/qu/unit#volt	Unit of measure for electromotive force (EMF), the electrical potential between two points. An electrical potential of 1volt will push 1ampere of current through a 1ohm resistive load.
wattPerSquareMetre	http://purl.oclc.org/NET/ssnx/qu/unit#wattPerSquareMetre	Rate of energy transfer per unit area equivalent to one watt transferred through a surface with an area of one square metre.

It is worth highlighting that all these elements have been imported into the M3-lite taxonomy, where we have harvested all the different quantity kinds and units from all FIESTA-IoT's first-party Testbeds and platforms (see Appendix II – "In-House" Testbed Quantity Kinds and Units). As can be inferred from these tables, there are number of elements that could not been mapped to any well-known ontology. In order to fill these gaps, the M3-lite taxonomy will create its own definitions and IRIs to "host" all these missing entities.

As long as the federation grows up, all new quantity kinds and units and units brought about by the different upcoming Testbeds will be fostered.

Last, but not least, with regard to those missing links that can be seen in the tables, they will be covered internally as legacy elements inside the M3-lite taxonomy. Said in other words, each missing IRI will be replaced by its corresponding M3-lite's own IRI.

2.2.10 IoT-O Ontology: A first approach towards unifying IoT Ontologies

IoT-O ontology is the first and innovative approach towards unifying existing IoT ontologies and being modular as depicted in Figure 12 [33]. IoT-O is a core-domain ontology designed for IoT to overcome heterogeneity and vertical silos in IoT. It is based on eight ontologies: Semantic Actuator Network (SAN)[24], DUL, Time, SSN, IoT-A, QUDT, MSM and hREST (see "Ontology Namespace" Table). The main novelty of this IoT-O ontology is the consideration of actuators since it is not addressed in SSN ontology or other ontologies. For this reason, the authors of [33] also designed Semantic Actuator Network (SAN) ontology and included it in the IoT-O ontology. IoT-O defines or reuses main concepts such as `ssn:Device`, `ssn:ObservationValue` for describing values measured by devices such as sensors and actuators, `qudt:QuantityKind` for describing units and `time:Instant`. They define relationships between IoT-Thing and services to explain that Internet Connected Objects (ICOs) can provide a service. On top of the ontology, they use Pellet to inference whether the system is defective, or to increase the luminosity of the bulb. They even share Semantic Web Rule Language (SWRL) rules on the web following the idea of the Sensor-based Linked Open Rules (S-LOR) [34]. Moreover, they are involved in the ETSI and oneM2M standardizations.

The ontology is well documented (based on LODE tool) and follows semantic web best practices, the NeOn methodology and Ontology Design Patterns (ODPs) recommendations. IoT-O ontology takes into considerations standardizations as well such as oneM2M, W3C SSN. The benefit of IoT-O is reusability based on modularization, ODPs, reuse of existing sources and alignment to upper ontologies.

However, they claim to design a modular ontology but do not explain how to select or use only one module/sub-part from the IoT-O ontology. They plan to use the IoT-O ontology to determine sensor or actuator failure. Thus, those issues refrain us from using the ontology within FIESTA-IoT.

[24] https://www.irit.fr/recherches/MELODI/ontologies/SAN

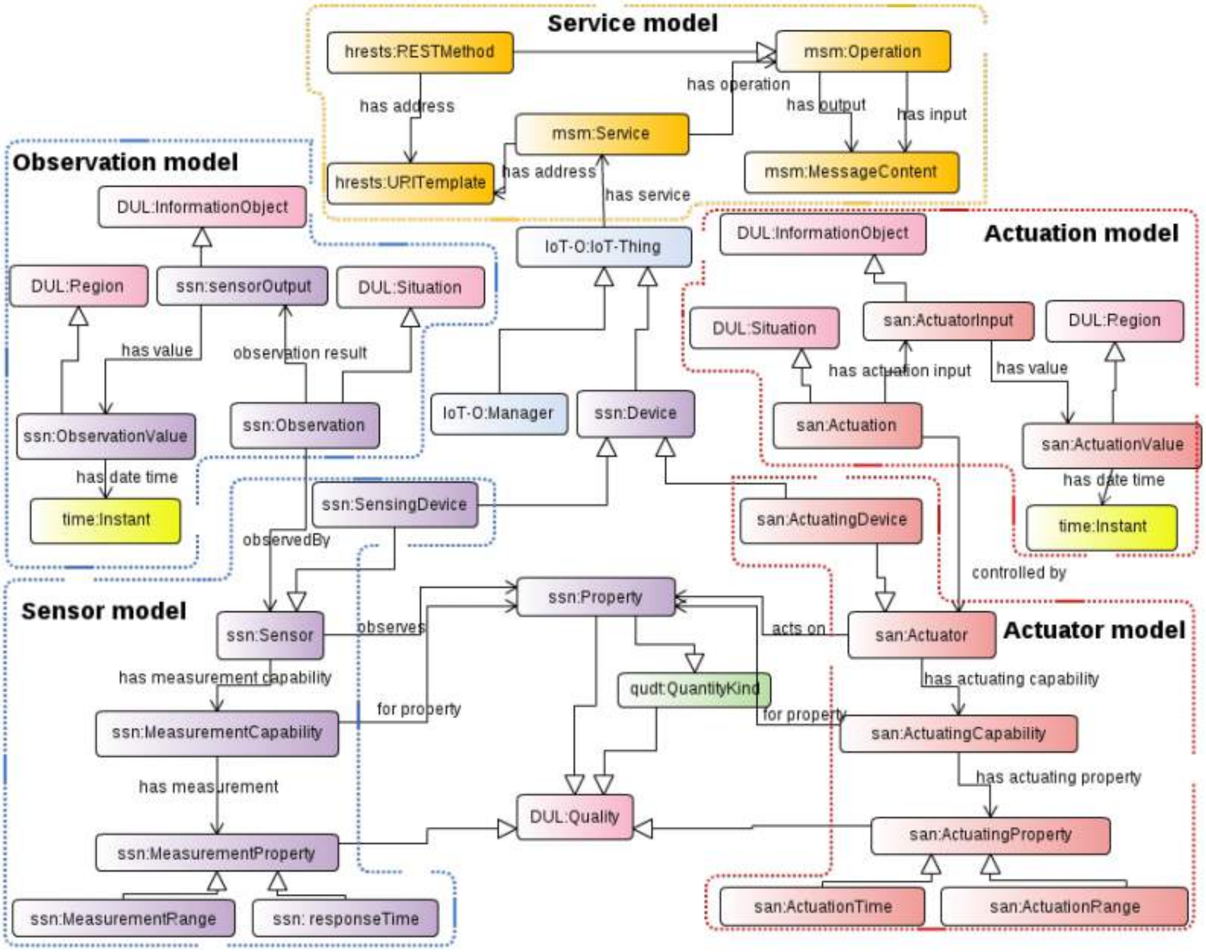


Figure 12: IoT-O Ontology [33]

2.2.11 oneM2M Base Ontology

oneM2M [35] is one of the main standardization bodies in IoT/M2M context, the ETSI M2M working Group. The oneM2M has already established the working group 5 Management, Abstraction and Semantics (MAS) in order to provide support by the system for application specific abstraction and semantics with regard to execution of M2M service.

In order to provide syntactic and semantic interoperability and to specify the semantics of resources, oneM2M constitutes the oneM2M Base Ontology. oneM2M uses OWL representation. It is expected that the systems utilizing the oneM2M ontology will define Sub-classes of some of the concepts within oneM2M in order to enable semantic interworking between oneM2M and non-oneM2M systems. The oneM2M base ontology consists of concepts like Thing, ThingProperty, Variable, MetaData, Device, Functionality, Aspect, command, Service, OutputDataPoint, InputDataPoint, Operation, OperationInput, OperationOutput, OperationStatus, AreaNetwork, InternetworkedDevice. These concepts are described as follows:

- **Thing**: Anything that is identifiable in oneM2M system is called a Thing. It may have association to object properties or to data properties (e.g. `hasThingRelation` or `hasThingProperty`). For example, a "room" is modelled as a Thing in oneM2M and could have a room temperature as a `ThingProperty` (via `hasThingProperty`) and could have a `hasThingRelation` "`isAdjacentTo`" to another room. In general "room" is considered a thing that is not able to communicate electronically. However, the sub-class of Thing that is able to interact electronically is a "Device".

- **ThingProperty**: denotes a property of a Thing. A Thing can be described with (the values of) it, but in general a Thing cannot influence that value or being influenced by it. A ThingProperty be can be retrieved or updated by an entity of the oneM2M System. For e.g., the indoor temperature of the room is considered as a value of a Thing "room", while the manufacturer can be considered a ThingProperty of a Thing "car". A ThingProperty of a Thing can describe a certain Aspect, e.g. the indoor temperature describes the Aspect "Temperature" that can be measured by a temperature sensor. A ThingProperty of a Thing also may have metadata.
- **Variable**: Variable is a super class to the classes: ThingProperty, OperationInput, OperationOutput, OperationState, InputDataPoint, OutputDataPoint. Member entities of this class may have some data (e.g. integers, text or structured data) that might be changed over time. These data usually describe some real-world Aspects (e.g. a temperature) and can have MetaData (e.g. units, precision). One sub-class is of Variable is SimpleTypeVariable which consists of Variables of simple XML types like `xsd:integer`, `xsd:string`, etc.
- **MetaData**: It contains data (like units, precision-ranges) about the Values of a Thing or about an Aspect. For example, the indoor temperature may have metadata: "Degrees Celsius".
- **Device**: A Device (sub-class of Thing) is a Thing that interacts electronically with its environment. A Device is dedicated to perform a particular task. A Device holds some logic and is producer and/or consumer of data that are exchanged via its Services in the network. It interacts through the DataPoints and/or Operations of its Services. In order to perform a task, the device performs one or more functionalities (Object Property: `hasFunctionality`) (class: Functionality). These functionalities are exposed as Services of the Device in the network. A Device can be formed of several (sub-) Devices. This uses ObjectProperty "`consistsOf`" (Device `consistsOf` Device). Each Device (or sub Device) needs to be addressable in the network.
- **Functionality**: The class Functionality describes the functionality necessary to accomplish or perform a task. A device can be designed to perform more than one Functionality. The class Functionality is defined to describe the human understandable meaning of what it "*does*". A Functionality observes or influences a certain Aspect. For example, a device "light switch" may have a Functionality "Controlling_ON_OFF" which refers to an Aspect "lighting" that is influenced by "light switch". This Class has two subclasses: ControllingFunctionality and MeasuringFunctionality. ControllingFunctionality only controls/influences real world Aspects where as MeasuringFunctionality only measures/senses real world Aspects.
- **Aspect**: An Aspect (a class) describes the real-world aspect that is related by a Functionality. It is also used to describe either the quality or the type of OperationInput or OperationOutput variables. The Aspect could be either an entity (physical or non-physical) or a quality.
- **Command**: A Command is defined to represent an action that should be performed to support a Functionality. A Command is exposed by an Operation to the network. Command can be parameterized by OperationInput and OperationOutput of the related Operation.
- **Service**: Service is a representation of a Functionality to a network which makes the Functionality discoverable, registerable, and remotely controllable in the network. One or more Functionalities can be represented by a Service. A Service is offered by a Device in the network.
- **OutputDataPoint**: An OutputDataPoint Class is a Variable of a Service. The Service is set by a Device that can update state information about the Service via RESTful API in its environment. The Device autonomously updates the OutputDataPoint (e.g. at periodic times). In order to enable a third party to retrieve the current value of an OutputDataPoint (out of schedule), devices may also offer a SET_OutputDataPoint Operation to trigger the device to update the data of the OutputDataPoint.

- **InputDataPoint**: An InputDataPoint is a Variable of a Service. It is set by Device in the same way as the OutputDataPoint in its environment. The Device reads out autonomously (e.g. at periodic times). In order to enable a third party to instruct the device to retrieve (out of schedule) the current value of a InputDataPoint devices may also offer a GET_InputDataPoint Operation to trigger the device to retrieve the data from the InputDataPoint
- **Operation**: A Command to network is represented by an Operation. It is the means of a Service to communicate in procedure-type manner. It can have OperationInput (data consumed by the Device) and OperationOutput (data produced by the device), as well as a method for describing how the Operation is invoked over the network. A Data Property of Operation "OperationState" indicates how the operation has progressed in the device. Two sub-classes of class Operation are defined in the base ontology:
    - **GET_InputDataPoint**: This class is a sub-class of Operation that will be offered by a Device in order to trigger the device to retrieve the data of an InputDataPoint (e.g. outside of the schedule when the device normally retrieves that DataPoint)
    - **SET_OutputDataPoint**: This class is a sub-class of Operation that may be offered by a Device to trigger the device to update the data of an OutputDataPoint (e.g. outside of the schedule when the device normally updates that DataPoint)
- **OperationInput**: The OperationInput Class represents the type of input of an Operation to a service of the device. It represents all possible values for that input. An Operation can have multiple OperationInputs and/or OperationOutputs. If an instance of an Operation is executed then the input value to that Operation must be an instance of its OperationInput classes.
- **OperationOutput**: The OperationOutput Class is represents the type of output of an Operation from a service of the device. This class represents all possible values for that OperationOutput. An Operation can have multiple OperationInputs and/or OperationOutputs. If an instance of an Operation is executed then the output values of that Operation are instances of its OperationOutput classes.
- **OperationState**: The OperationState Class represents the current state during the lifetime of an Operation. It describes all possible values for that state (enumerated individuals). The OperationState is set during the progress of the operation by the entity invoking the operation. It takes values like "data_received_by_application", "operation_ended", "operation_failed" or "data_transmitted_to_interworked_device".
- **AreaNetwork**: The AreaNetwork Class is characterized such as physical properties (e.g. IEEE_802_15_4_2003_4GHz), a communication protocol (e.g. Zigbee_1_0) and potentially a profile (e.g. Zigbee_HA)
- **Internetworked Device**: This is a part of an AreaNetwork. It is not an oneM2M Device. It can be only accessed from the oneM2M System by communicating with a "proxied" (virtual) device that has been created by an Interworking Proxy Entity (IPE). The InterworkedDevice class describes the "proxied" (virtual) device that is represented in the oneM2M System as an individual Application Entity (AE) resource or a child resource of the AE of its IPE.

We describe the above concepts as we aim to align FIESTA-IoT ontology with oneM2M concepts.

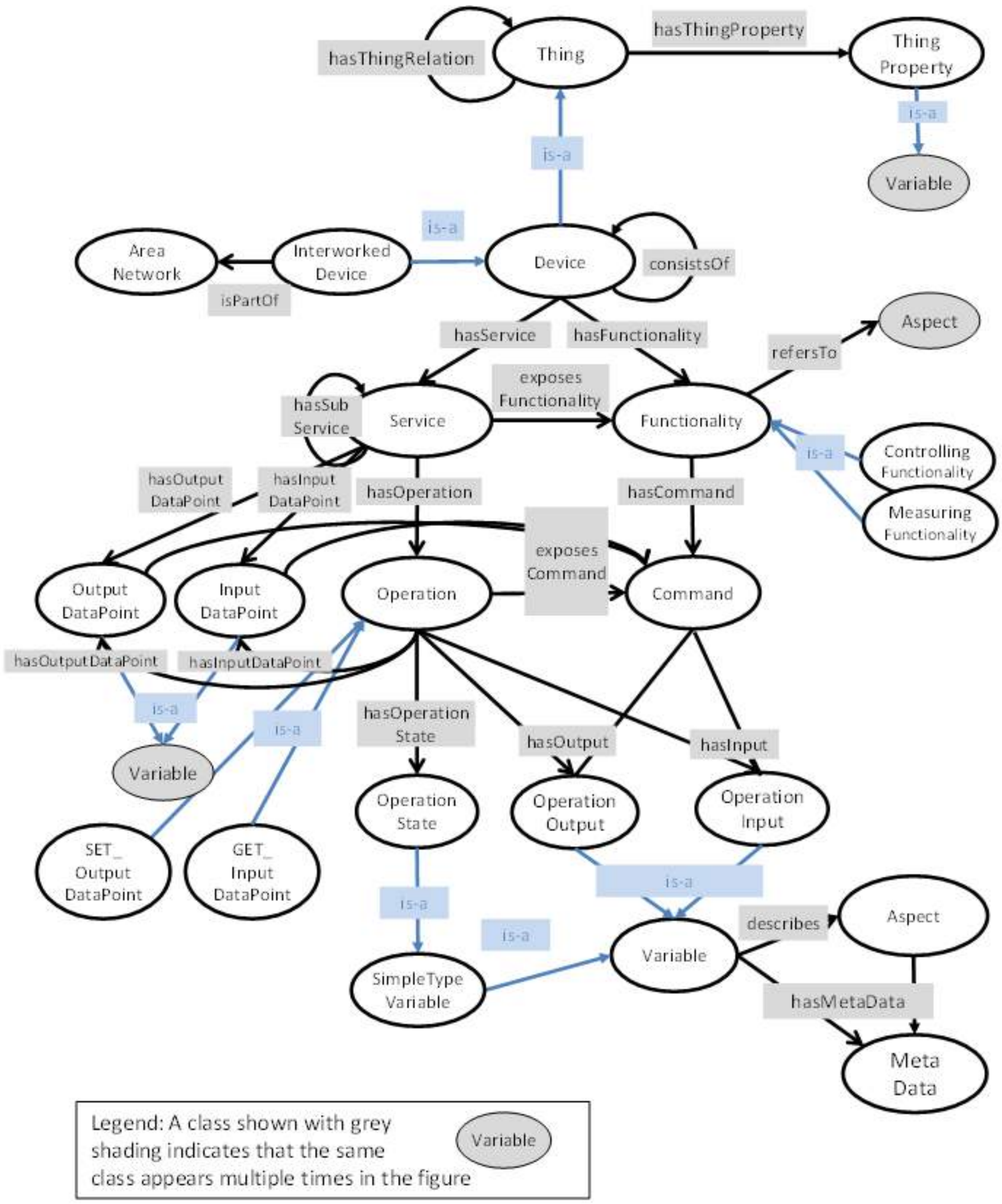


Figure 13: oneM2M base ontology [35]

Figure 13 shows the necessary classes and properties within oneM2M Base Ontology. In Figure 13, classes are denoted in oval and Object Properties are denoted by black arrows. Sub-Class relations are denoted by blue arrow.

The oneM2M standardization including the Base Ontology will be used in diverse IoT service domains (such as traffic, healthcare, and smart city) in order to realize them. Furthermore, these IoT services will be interworked with FIESTA-IoT according to the main goal and objectives of the FIESTA-IoT project. In order to provide interoperability between such IoT services and FIESTA-IoT, the oneM2M Base Ontology will be become one of key approach. Accordingly, the oneM2M Base Ontology and FIESTA-IoT will have a very close relationship.

2.2.12 SAREF Ontology

The Smart Appliances REFerence[25],[26] (SAREF) [36] ontology is designed for household and home appliances in residential buildings, especially for the purpose of energy management. SAREF aims to align existing ontologies in the domain of smart appliances. Many standards have been proposed to enable the interoperation of appliances from diverse vendors. However, the number of standards is so high that overlapping is inevitable. To address this problem, the European Commission launched a study for the purpose of proposing a reference ontology gathering the efforts of existing appliances standards relevant for energy efficiency. The final result of this study is the SAREF reference ontology that is intended to be transferred to European Telecommunications Standards Institute (ETSI) Smart Machine to Machine (SmartM2M) that could contribute it to International Machine-to-Machine Standardization (oneM2M) initiative.

The central concept of SAREF is the "device" (see Figure 14). Devices are tangible objects that are designed to accomplish one or more "functions". Basic functions, such as "switch on/off", are provided in SAREF and can be combined to have more complex functions in a single device. A function is associated with "commands", for example, the function "switching on/off" has commands "switch on" and "switch off" which are atomic actions performed by an actuator. The device has one or more "states" which can be changed according to command. "Service" is one or a set of functions that can be registered, discovered and used by other devices on the network. A device offers a service.

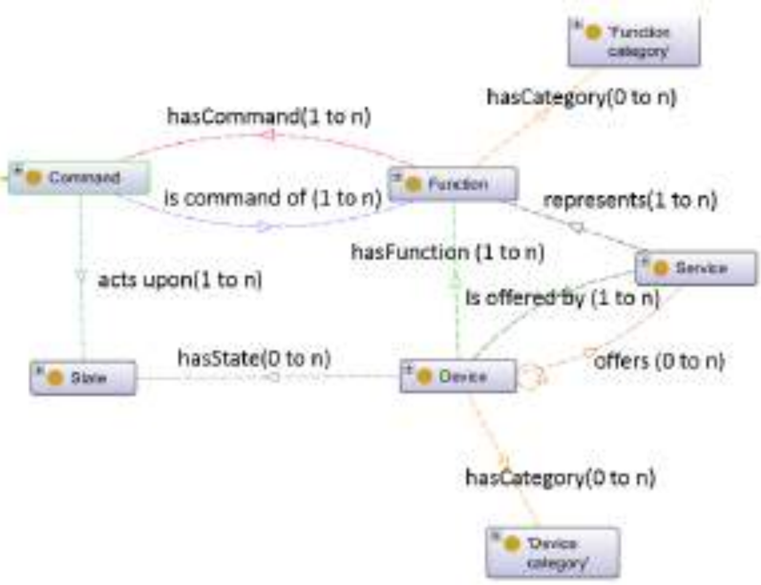


Figure 14: Key Concepts in SAREF Ontology

"Function/device category" provides a way of classifying functions/devices according to some point of view. In SAREF, the classification of Device is "flat". This means, instead of sub-classing a device subclass, such as "Sensor", for specific types, subclass of device is created and related to other subclasses by "`hasCategory`" relation. Thus, there is no hierarchy in the Device class. Let's take a concrete example. In a classic taxonomy, a humidity sensor and a temperature sensor are both subclass of the class of sensor that is a child of Device (see Figure 15). The levels of hierarchy created in this way can grow quickly as well as the number of issues. For example, if a "humidity temperature" sensor is created as the subclass of both temperature and humidity sensor, the multiple inheritance issue

[25] https://sites.google.com/site/smartappliancesproject/ontologies/oma-lightweight_m2m-ontology
[26] http://ontology.tno.nl/saref/

occurs. More the levels of hierarchy, harder the maintenance is. The “flat” categorization is more flexible and no hierarchy needs to be maintained (see Figure 16).

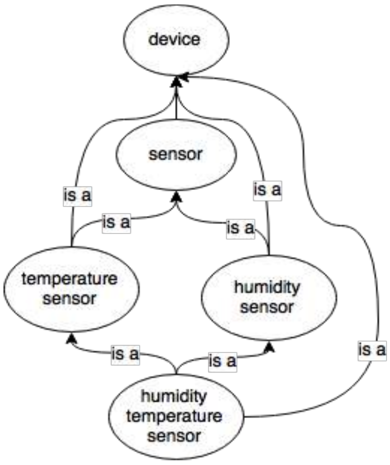


Figure 15: Device category using subclass

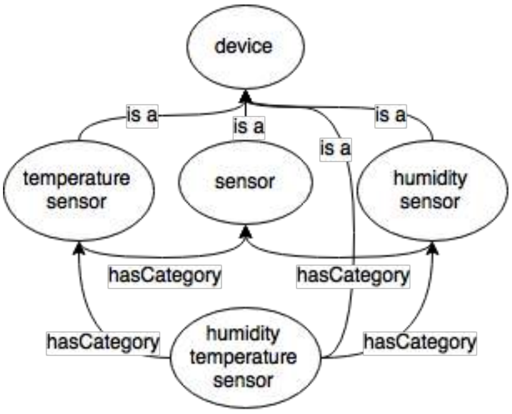


Figure 16: Device category using "hasCategory" relation

A mapping is done between the SAREF reference ontology and other ontologies, such as W3C SSN [12], UPnP [37] is also done. As an on-going effort by SAREF reference ontology authors, the mapping of oneM2M base ontology and the present SAREF reference ontology is being done. This would allow us, in future, to look into to map SAREF reference ontology to the FIESTA-IoT ontology.

2.2.13 Open-Multinet Ontology

Within the FIESTA-IoT project it is envisioned to register involved Testbeds and add meta-information about the infrastructures and sensor information they are offering. The purpose of the OWL-encoded Open-Multinet (OMN) [38] ontology is to build a formal model to describe federations of infrastructures and the whole life cycle of a services and resources on a semantic level. While initially developed to act as a canonical model to solve compatibility issues between federated Testbeds and enable software tools to understand the meaning of resource descriptions, it can further be used as the basis for a knowledge-base about federated infrastructures as demonstrated in DBcloud[27] [39]. This allows the use of reasoners and formal concepts to query, link, chain, map, combine and validate information from different infrastructures.

[27] http://lod.fed4fire.eu

To broaden its scope beyond experimentation in federated Testbeds and with the goal of defining a standard, the W3C Federated Infrastructures Community Group [28] was established. Using the OMN ontologies as the starting point, a larger community is validating their adaptability for further fields of application, such as Intercloud computing, in order to further refine the models and to incorporate additional requirements. As a result, the work is discussed within the IEEE Standard for Intercloud Interoperability and Federation (P2302) [40], [41] working group.

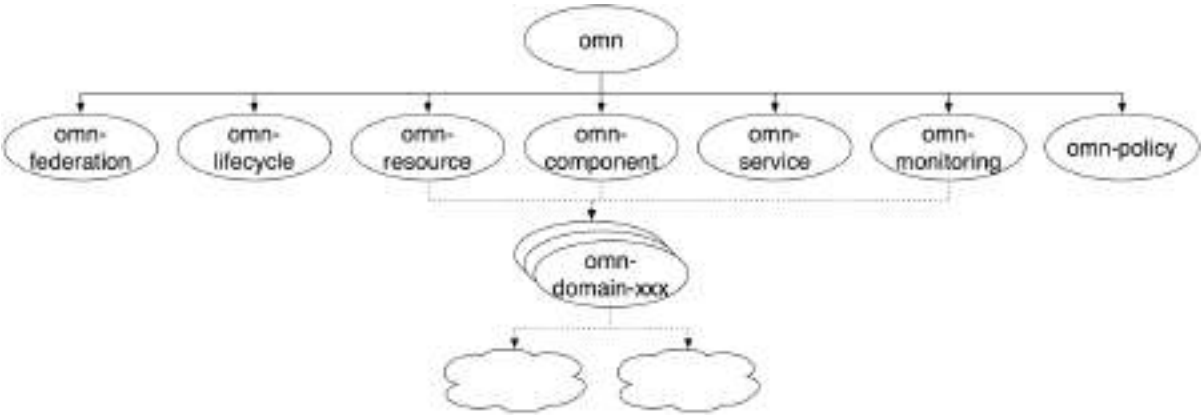


Figure 17: OMN upper ontologies [38]

As the definition of such an upper ontology set for the broad application context of federated infrastructures requires the involvement of many stakeholders, the vocabulary was developed in close collaboration with a community of relevant experts within the FIRE [42] and Global Environment for Network Innovations (GENI) [43] community and authors of related ontologies. The ontology bundle is split into a hierarchy of a number of different ontologies as shown in Figure 17. The OMN ontology on the highest level defines basic concepts and properties, which are then reused and specialized in the subjacent ontologies. Included at every level are (i) axioms, such as the disjointedness of each class to allow proper reasoning; (ii) links to concepts in existing ontologies, such as the Network Mark-Up Language (NML) [44], the Infrastructure and Network Description Language (INDL) [45], [46] and the Networking innovations Over Virtualized Infrastructures (NOVI) [47] ontology (see Figure 18); and (iii) properties that have been shown to be needed in 6 related ontologies.

The OMN upper ontology defines the abstract terms required for describing federated infrastructures in general. Figure 19 illustrates an overview of the key concepts and properties. The main concepts are as follows based on [38]:

- `Resource`: A stand-alone component of the infrastructure such as a network node, which can be provisioned, i.e., granted to an experimenter.
- `Service`: A manageable entity, which can be controlled and/or used via APIs or capabilities it supports, e.g. an SSH login.
- `Component`: A part of a Resource or a Service, e.g. a port of a network node.
- `Attribute`: Description of the characteristics and properties of a specific Resource, Group, or Component, e.g. Quality of Service (QoS).
- `Group`: A collection of resources and services, e.g. a Testbed or a requested network topology.
- `Dependency`: A unidirectional relationship between `Resource`, `Service`, `Component` or `Group`, which opens up the possibility to add more properties to a dependency via annotation.
- `Layer`: A place within a hierarchy that a specific Group, Resource, Service or Component can adapt to.

[28] https://w3.org/community/omn

- `Environment`: The conditions under which a Resource, Group, or Service is operating, for example, concurrent virtual machines.
- `Reservation`: A specification of a guarantee for a certain duration, which is a subclass of the Interval class of the W3C Time ontology [48], used to encode, for example, start and end times.

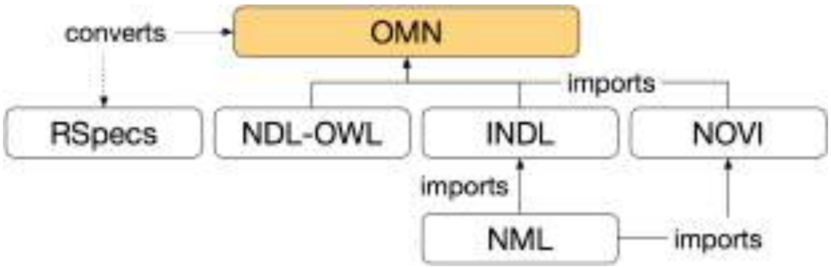


Figure 18: Relation between OMN and other work [38]

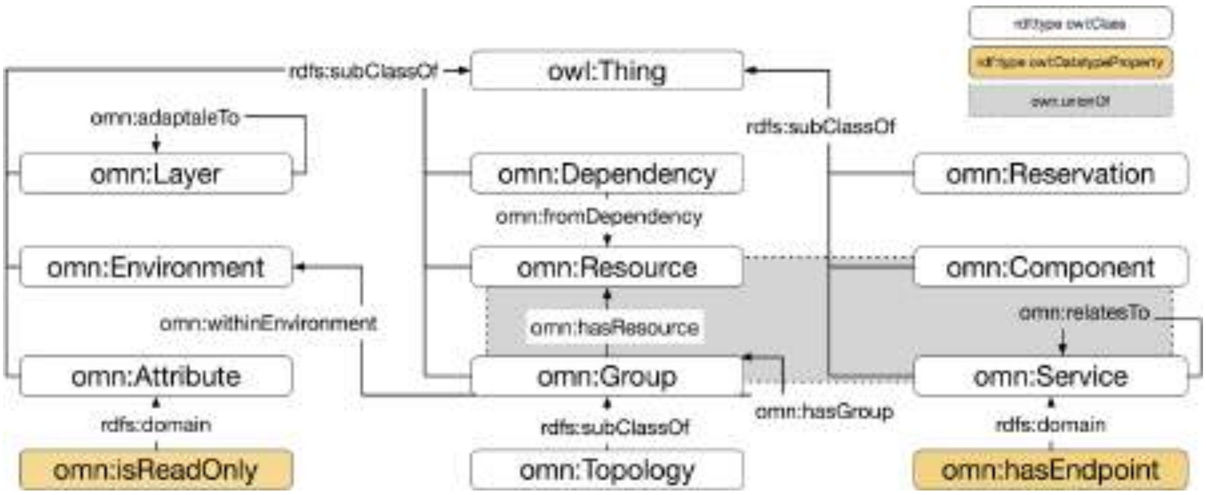


Figure 19: Key concepts and properties of the OMN upper ontology [38]

Besides the main concepts, 21 properties have currently been defined, including inverse counterparts in order to support rich querying and inferencing.

- `adaptableTo` relates a Layer to another Layer to which it can adapt (e.g. from Ethernet to Internet Protocol (IP)). The inverse is `adaptableFrom`.
- `adaptsFrom` determines the `Group`, `Resource`, `Service` or `Component` from which another `Group`, `Resource`, `Service` or `Component` adapts. The inverse is `adaptsTo`.
- `fromDependency` relates a `Group`, `Resource`, `Service` or `Component` to the `Dependency` to which it belongs. The inverse is `toDependency`.
- `relatesTo` generally relates a `Component`, `Resource`, `Service` or `Group` to another `Component`, `Resource`, `Service` or `Group` to which it belongs. The subproperty `dependsOn` claims a direct dependency.
- `hasAttribute` the Attribute associated with a `Component`, `Resource`, `Service` or `Group`. The inverse is `isAttributeOf`.
- `hasComponent` links a `Component`, `Resource`, or `Service` to its subcomponent. The inverse is `isComponentOf`.
- `hasGroup` connects a `Group` to its subgroup. The inverse is `isGroupOf`.
- `hasReservation` relates `Group`, `Resource`, or `Service` to its `Reservation`. The inverse is `isReservationOf`.
- `hasResource` declares that a specific `Group` has a `Resource`. The inverse is `isResourceOf`.

- `hasService` declares that a `Group`, `Resource` or `Service` provides a `Service`. The inverse is `isServiceOf`.
- `withinEnvironment` defines the `Environment` in which a `Group`, `Resource`, `Service` or `Component` operates.

2.2.14 Various other IoT-related Ontologies

Various other ontologies in the field of SSN exist. Some of them are designed for project specific purposes. These include ontology used in CityPulse, IoT.est, LSM, Read4SmartCities, STAR-CITY, OBOE, SWAMO, etc. Further, many other ontologies also exist, e.g., SemSOS, SPITFIRE. Some other ontologies like WSG84, GeoNames, GoodRelations are linked by these ontologies or can be linked against them. We are exploring the possibility of including and aligning FIESTA-IoT Ontology with some of the ontologies mentioned in this section (Section 2.2) in the next version of this document.

2.3 Ontology Annotation tools

In this section we provide related available annotation tools.

2.3.1 SAOPY

SAOPY[29] is an annotation tool to annotate stream data following the SAO[8] [25] ontology. It provides ontology-dependent API for annotating data following the SAO ontology. As the API strictly follows the ontology structure, it becomes hardly possible to make mistakes while annotating the data. Although by default, annotation would require human interaction for creating instances of sensor devices since the interaction is through command prompts, a script can be useful. In FIESTA-IoT, SAOPY will be investigated further in order to support semi-automated annotation that complies with the FIESTA-IoT ontology.

2.3.2 Sense2Web: The linked Sensor Data Platform

Sense2Web[30] is a web-tool (see Figure 20, snapshot of some relevant functionalities is provided via Figure 22 - Figure 24) and API that can be used to do registration of Virtual Entities (VEs), Resources and Services. It is compliant with the IoT-A ontology and is planned to be migrated to IoT-lite. It also offers simple search capabilities and enables SPARQL queries.

Sense2Web provides a semantic annotation framework to describe sensor devices and services. The semantically annotated sensor data is published as linked RDF data and then made available to other applications via SPARQL endpoints. The annotations describe the sensor device/resource according to a semantic model that described in [49]. Sense2Web also provides a mash-up application using the published data and open data resource on the Web to demonstrate reasoning and interpretation of linked sensor data and resource descriptions. Figure 21 shows the user interface for publishing a new sensor description in Sense2Web. It uses Jena API[31] to query DBpedia and GeoNames knowledge bases to enrich the description of the sensor devices/services with metadata. Figure 21 shows suggestions that are retrieved from DBpedia for a sample query, “Guildford”. The submitted variables are stored in RDF/XML format. The semantic annotation has been performed in early versions using Extensible Stylesheet Language Transformations (XSLT). In the later versions this has been replaced by a customized mapping function using the Jena API. The tool has evolved to adopt the last version of the information model defined in IoT-A. It has

[29] http://iot.ee.surrey.ac.uk/citypulse/ontologies/sao/saopy.html
[30] http://iot.ee.surrey.ac.uk/s2w/
[31] https://jena.apache.org/documentation/rdf/

been incorporated in the FIWARE platform as a standalone tool in the IoT Discovery Generic Enabler.

In FIESTA-IoT, Sense2Web will be adopted as a base asset, modified and developed further to support the FIESTA-IoT ontology and also expose a more RESTful interface, addressing issues relating to (but not limited to) URL structure, content negotiation and dereferenceability of resources.

Figure 20: The linked sensor data platform Tool.

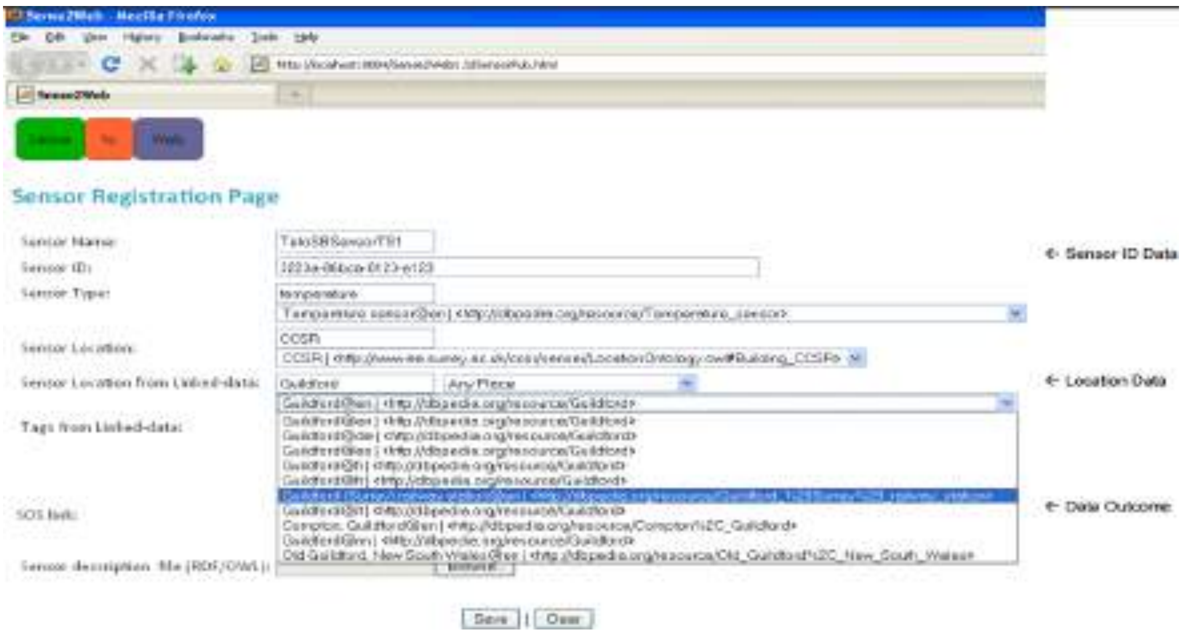


Figure 21: Sense2Web: Early UI for publishing sensor descriptions

Figure 22: Register an IoT resource form page in Sense2Web platform

Figure 23: Object Map overlay within Sense2Web platform

Figure 24: Query an IoT Description

2.4 Ontology Validation tools

In this subsection we provide various available ontology validation tools.

2.4.1 SSN Validator

The SSN validator[32] is a web application (see Figure 25) that allows ontology developers to validate their data that are based on the SSN ontology. To validate, users can input their data either by uploading a file or directly inputting the data in the textbox. An SSN instance "smart-knife" is provided as an example.

The validator service receives the data and compares it against the SSN ontology and other ontologies that are frequently associated with it. The result given is a validation report that reports any inconsistencies with the data. The application also provides a tag cloud that illustrates the popularity of the terms that are used by all the data that has been submitted for validation.

When a document is submitted to the service, the data is passed to the validator that uses the Jena Eyeball API. The document is then parsed via a SPARQL engine whereby the terms it contains are extracted and stored, and a validation report is generated (see Figure 26).

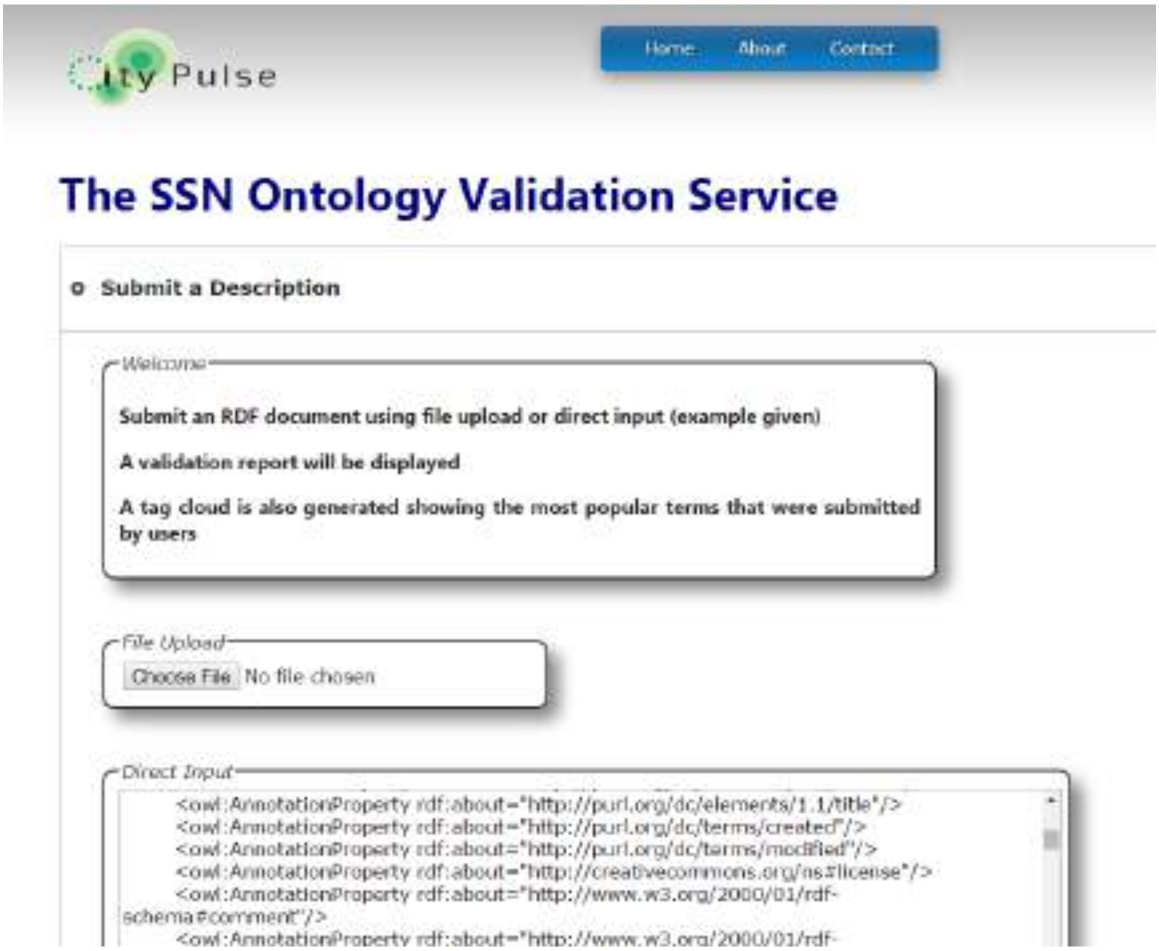


Figure 25: SSN Validator Tool.

[32] http://iot.ee.surrey.ac.uk/SSNValidation/

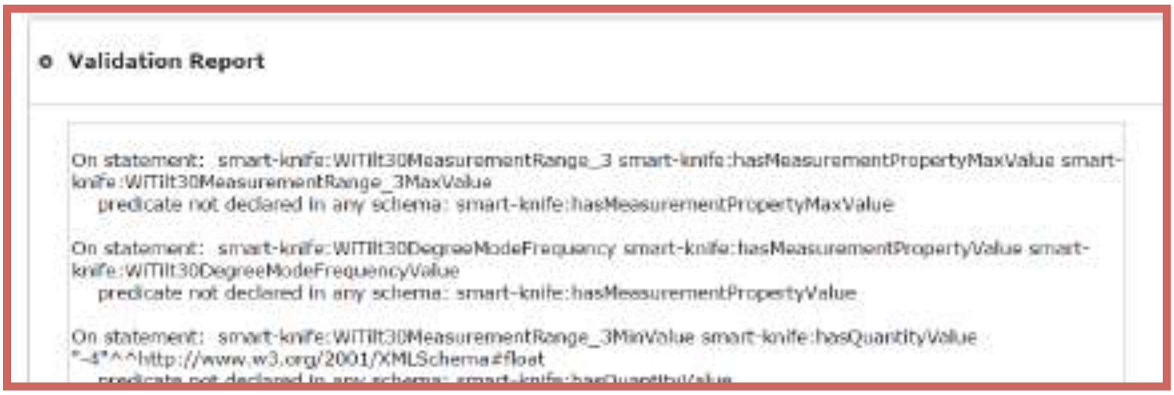


Figure 26: SSN Validator Tool. Results Window

2.5 Creation of Composite Data/Services

Creation of composite Data/Services is a concept where aggregation of data or services is done to achieve a common objective. Service composition would enable combining different services to obtain a more sophisticated service. Since semantic web technologies are employed within this project, an intuitive way is to employ Semantic Web Services [50]. For instance, having two simple services such "Service 1: Get Average Temperature" and "Service 2: Get Food Suggestions" would enable a more sophisticated services "Service 3: Get Food Suggestions according to the temperature", by combining previous simple services.

In FIESTA-IoT as the focus is on the data, creation of composite data/services can be done in following ways:

- **Reasoning or inferences**: data composition can be done using reasoning or inferences on the semantically annotated data stored in the data repository. The reasoning or inferences often produce new knowledge based on the existing gathered data. The gathered inference can be generated based on rules that are executed over the available data. These rules can be either simple rules that involves data for single type of QK or can be very complex rules involving data from multiple types of QK. However, generation of such rules and examples of such rules is a task to be performed in T3.4. The results of T3.4 are presented in Deliverable "D3.4: Concept and Development for IoT data analytics and IoT stream and service management". Nevertheless, an example of simple reasoning is: consider temperature data, tag like "Extreme", "Freezing", and "Hot" can be generated based on temperature values. Further, an example of complex reasoning using available data from various sensors is: consider temperature data and humidity data, tags like "Rain" and "Dry" can be generated. Nevertheless, inferences can also be made on location data that accompanies each measurement. From location data, mobility patterns and related properties can be inferred [51]. Some mobility characteristics that can be inferred with the available data are regular trips, accompanying details, personal places and travel frequencies (all well-known properties in the study of human mobility). However, such inferences can also be generated by various actors dealing with FIESTA-IoT platform (such as Raw-data producers, Virtualizers, Knowledge producers and Experimenters [1]) and made available to FIESTA-IoT platform. Note that Testbeds could take up the role of Raw-data producers, Virtualizers and Knowledge producers, and already produce such knowledge and provide such inferences along with the raw data. While Experimenters could utilize FIESTA-IoT data, produce such inferences, and then provide it back to FIESTA-IoT. This will require the FIESTA-IoT ontology to be capable enough to semantically annotate the inferred data.
- **Virtual Entities (VEs):** FIESTA-IoT architecture [1] allows the creation and storage of semantically annotated data produced by VEs. VEs such as a building, which embraces

many resources inside can be generated. Provision of such virtual entities via service is provided by FIESTA-IoT platform. Testbeds and Experimenters registered with FIESTA-IoT platform produce VEs. For Class-I Testbeds, the VEs are stored in the VE repository at the Testbed. However for Class-II and Class-III Testbeds the VEs should be created within the FIESTA-IoT platform. Experimenters can also create their own VEs and utilize them in their experiments.

- **Experiments:** FIESTA-IoT aims to provide Experimentation-as-a-Service (EaaS). Experiments can be simple as well as complex. Experiments stored within FIESTA-IoT platform follow a specific workflow. Already stored experiments and their workflow integrated with the aim of EaaS, provide a way to perform composition of data/service. The generation of experiments is done via WP5. More description about existing experiments within FIESTA-IoT platform is available in [52] where three in-house experiments are defined.

Given above ways, we intend to continue this study to identify more aspects involved in creation of composite data/services.

3 ALIGNING EXISTING ONTOLOGIES

The design of the FIESTA-IoT ontologies takes inspiration from the following:

- Modularization encourages the design of ontologies in separated modules to make them easier to reuse and/or extend.
- Ontology Design Patterns (ODPs) [53].
- Reuse of existing sources [54]
- Alignment to existing ontologies
- Ontology methodologies such as NeOn [55]**.**
- Semantic web best practices applied to IoT as stated in [9]

In the next subsections, we present methodology followed, evaluation of the various models present, ontologies aligned and the FIESTA-IoT ontology.

3.1 Methodology followed to build Ontology

Noy et al. explain in the second step of their **ontology development 101 methodology** that ontology designers should consider reusing existing domain knowledge (e.g., ontologies) [56].

Corcho et al. survey the existing ontology methodologies [57] and answer the following questions:

- Which methods and methodologies can I use for building ontologies?
- Which tools give support to the ontology development process?
- Which languages should I use to implement my ontology?

The **NeOn** project[33] recommends reusing available knowledge and proposes a set of methodologies [55]. The NeOn project focuses on nine scenarios: (1) from specification to implementation, (2) reusing and re-engineering non ontological resources, (3) reusing ontological resources, (4) reusing and re-engineering ontological resources, (5) reusing and merging ontological resources: ontology matching tools enable ontology aligning or merging, (6) reusing merging, and re-engineering ontological resources, (7) reusing ontology design pattern (ODPs), (8) restructuring ontological resources, and (9) localizing ontological

[33] http://neon-toolkit.org/wiki/Main_Page.html

resources to translate of all the ontology terms into another natural language. We are mainly interested in the scenario 3 to help IoT developers in reusing ontologies relevant for IoT. The others future steps are interesting for re-designing ontologies in an interoperable manner and not reinventing the wheel at each ontology development' to speed up the ontology development process.

On-to-Knowledge is another methodology for designing ontologies comprised of four steps: (1) kick-off, (2) refinement, (3) evaluation, and (4) ontology maintenance [58]. Such methodologies should be followed to reuse or design ontologies and datasets.

The methodologies provided by the NeOn project [55] and Noy et al. [56] are the most popular ones. These methodologies will be followed within FIESTA-IoT, since the main purpose is dealing with semantic interoperability. All of these methodologies promote the reuse ontologies to encourage interoperability.

Following the above approach, the methodology followed by us in order to create the ontology is:

- **Gather requirement** with those coming from IoT-A ARM and its concepts and from the in-house Testbeds and Experimenters [59]
- **Study existing Semantic Models and Taxonomies** related to IoT and identify shortcomings.
- **Merge** relevant and necessary Semantic Models and **create instances** along with **reference annotation tools**
- **Document** Ontology, Taxonomy and Annotation Tools.
- **Extend and update.**

3.2 Ontologies Aligned

Many of the existing ontologies are not well connected and are domain specific (See section 2.2). In the Section 2.2, we identified the usefulness of the ontology and identified if the ontology was validated with best practices. We found that only M3 ontology was validated[34] and, in most of the IoT related ontologies Semantic Web practices are not followed. Further, the reason for the ontologies not being widely adopted as foreseen is because they add processing time, especially when dealing with large-scale data as the IoT streams.

Our vision of the ontologies is to make it lightweight, and easily to understand and adopt. FIESTA-IoT ontology is based on IoT-lite, which is a lightweight ontology for describing IoT resources and services (see Section 2.2.3). IoT-lite could be extended with different modules to further describe more detailed concepts of the resources and data. IoT-lite is linked to M3-lite (see Section 3.3.4) which is the taxonomy of quantities and units. M3-lite's hierarchical approach enhances the definition and hierarchical relationship between quantities and units. For description of sensory stream data we will use SAO ontology that allows the annotation of stream data as raw data or aggregate data. Currently, we are looking into extending FIESTA-IoT ontology to describe Experiments.

3.3 FIESTA-IoT Ontology

In this section we present the FIESTA-IoT ontology by describing the core concepts and issues addressed, following with the overall vision of the ontology that works along with the M3-lite taxonomy.

[34] http://www.sensormeasurement.appspot.com/?p=ontologies#home

3.3.1 Core concepts

As the project has adopted the IoT-A Architecture Reference Model (ARM) as a template for the FIESTA-IoT architecture [1], so also do its core concepts. These are the Resource, the Virtual Entity and the IoT Service. According to IoT-A terminology [60]:

- 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.”

As mentioned in the Section 2.2.2, IoT-A has specified an information model that describes these concepts, but the issue is with the model’s interoperability, and complexity. In FIESTA-IoT, the aim is to address and maximize interoperability as much as possible.

3.3.2 IoT-lite

IoT-A lacks the adoption of other well-known ontologies that cover fundamental properties, e.g. those ones relating to space and time. Other concerns to note in IoT-A ontology are that the Entity sub-model seems to be a replication and extension of the Resource Model, and hence questions about the redundancy of properties arises, where one sub-model would have been sufficient.

In FIESTA-IoT, the Resources are mainly related to Sensor, Actuator or Tag hosting devices. From the analysis of the ontologies that relate to IoT, SSN stands out by far as a well-adopted ontology, and hence should serve as the base of an interoperable ontology. The conflict between SSN 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 “Process”. In IoT-A, this property is used in a different context, in the 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 can play an important role in the FIESTA-IoT ontology. The Device concept can be adopted so that a FIESTA-IoT Resource is hosted on a `ssn:Device`. This *could* be made using an explicit property e.g. “`isHostedOn`”. In this case, we need to define a “Resource” concept although this concept will not provide any added value to the information, especially upon querying it. Also to note in IoT-A, multiple Resources can be hosted on a single Device. Similarly 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 can be made, whereby one Resource is hosted on one Device and hence can be 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 has other aspects such as actuation and identification. Therefore, like for the IoT-A Resource, the SSN Device should have other instantiations for Actuators and Tags. This is where the IoT-lite ontology can play an essential role in extending SSN to include these concepts.

3.3.3 Ontology

The current version of the FIESTA-IoT Ontology (see Figure 27) is a merge of existing IoT ontologies into a single one. As can be seen in the Figure 27, it fosters concepts from a

number of “third-party” ontologies such as WGS84[35], W3C SSN, IoT-lite, M3-lite Taxonomy, DUL and QU.

Summing up the different classes, object properties and data properties that can be observed in Figure 27, we highlight the following elements:

- Technically speaking, the root of the graph for every resource would be its actual owner (i.e. the Testbed). Namely, a `ssn:Deployment` individual will reflect this role.
- Based on the GEO ontology, we describe the physical location of the devices. For that, we used the `geo:lat` and `geo:long` object properties. A similar approach to this geo localization is followed with the Coverage class and all its underlying subclasses (e.g. Polygon, Circle, Rectangle, etc.), located at the right side of the figure. This latter property will be utilized in elements (e.g. Virtual Entities) where the point localization is just not enough.
- a Platform describes “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” [12]. Upon this description, we will attach the physical location of each device to an instance of a Platform class.
- Device (based on `ssn:Device`), the core of the resource description. According to the SSN definition, a Device 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)”. In the scope of FIESTA-IoT, these smaller devices might be ActuatingDevices, TagDevices or SensingDevices. From now on we are going to focus on this later case.
- Directly connected to these Devices we have the class Service (or IoT service), which are the elements that expose the resources/devices. It is worth highlight that, since all the types of Devices actually inherit its properties, an IoT service might indistinctly apply for Devices, SensingDevices, etc. Besides, as will be seen in Section 4.1, these entities are embedded as a part of the resource description.
- 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 will create associations with potential number resources through the observed “Attributes”, which will define the VE properties.
- A SensingDevice (taken from `ssn:SensingDevice`: “A sensing device is a device that implements sensing”). Said in other words, this will represent the physical sensors deployed throughout the different Testbeds that will shape the FIESTA-IoT federation. As can be appreciated from the figure, Sensor is a supperclass of SensingDevices and will play an essential role, both in describing the different resources and services, and handling the data gathered from the measurements (bottom part of the figure).
- To map a SensingDevice/Sensor to the physical phenomena that it is actually “sensing”, we can find an explicit association to the M3-lite taxonomy, which will be thoroughly described in 3.3.4.
- Apart from the physical aspects that shape the main attributes of SensingDevices, it is also important to include some information that introduces another type of sensible data (i.e. metadata), such as the frequency of the measurements, the accuracy of the sensors, the precision and more information that might come up in future versions of the ontology.
- Finally, the lower part of the figure represents part of the graph in charge of annotating the measurements generated by the SensingDevices/Sensors. As it can be seen, there

[35] WGS84 is actually a basic RDF vocabulary that provides the Semantic Web Community with a namespace for the representation of latitude, longitude and other information about spatially-located things. For reference please see https://www.w3.org/2003/01/geo/

are five parts that shape a measurement/observation: timestamp, location[36], the actual value of the measurement itself (linked again to the M3-lite taxonomy) and the corresponding tuple QuantityKind/Unit. We will deepen into the details of these elements in Section 4.1.

For the sake of having an illustrative example, we encourage the reader to take a look at Section 4.1 and Appendix III – Sample Usage of Ontology, where we have put different arbitrary annotations of either resources or measurements, gathered from the various "in-house" Testbeds.

One of the things that have not been brought up so far is the linkage between the tangible outcomes of this ontology, i.e., the annotated descriptions/documents and the Functional Components that will be in charge of their storage/management. For further details about the FIESTA-IoT Functional View and the system use cases that might affect to each of these descriptions, the reader should refer to [1]. Namely, the current version of the ontology yields up to three different graphs: one for resources (including the IoT Services as part of them), another for Virtual Entities (whose VE properties might embrace a number of resources) and the last one, for those measurements/observations generated by the various devices that are part of the whole FIESTA-IoT ecosystem. In terms of management, the IoT Service/Resource Registry, the Virtual Entity Registry and the Meta-Cloud Data Endpoint, respectively, are in charge of their management and storage. As for this last task, each component will handle its own Triple Store repository, based on the Apache Jena framework[37], to both record the registered resources/services, VE entities (created by the virtualizers) and all the measurements injected from the underlying Testbeds, respectively.

Last, but not least, another point that is worth emphasizing is the decision of providing dereferenceable links in a number of entities, so as to introduce the Web of Things[38] paradigm to the extent of the FIESTA-IoT ontology approach, thus they would become accessible through standard Web protocols. Even though, technologically speaking, it would not be a problem to include a deferenceable link into every entity of the graph, the reality is that in many cases it would be pointless, since the information behind some of them would be meaningless by their own. Hence, a decision was taken about the usage of dereferenceable links only for those entities that have the possibility to expose a service, that is to say, a Device or a SensingDevice. In addition, the own nature (by definition) of IoT services make them "linkables" through any standard web application, since their inherent data type is no other but a URL defined by regular IoT service endpoints.

[36] This will be extremely useful in mobile sensors, where the location might vary between consecutive measurements. To keep the homogeneity, static nodes will also transmit their position in every observation.

[37] https://jena.apache.org/

[38] http://webofthings.org/

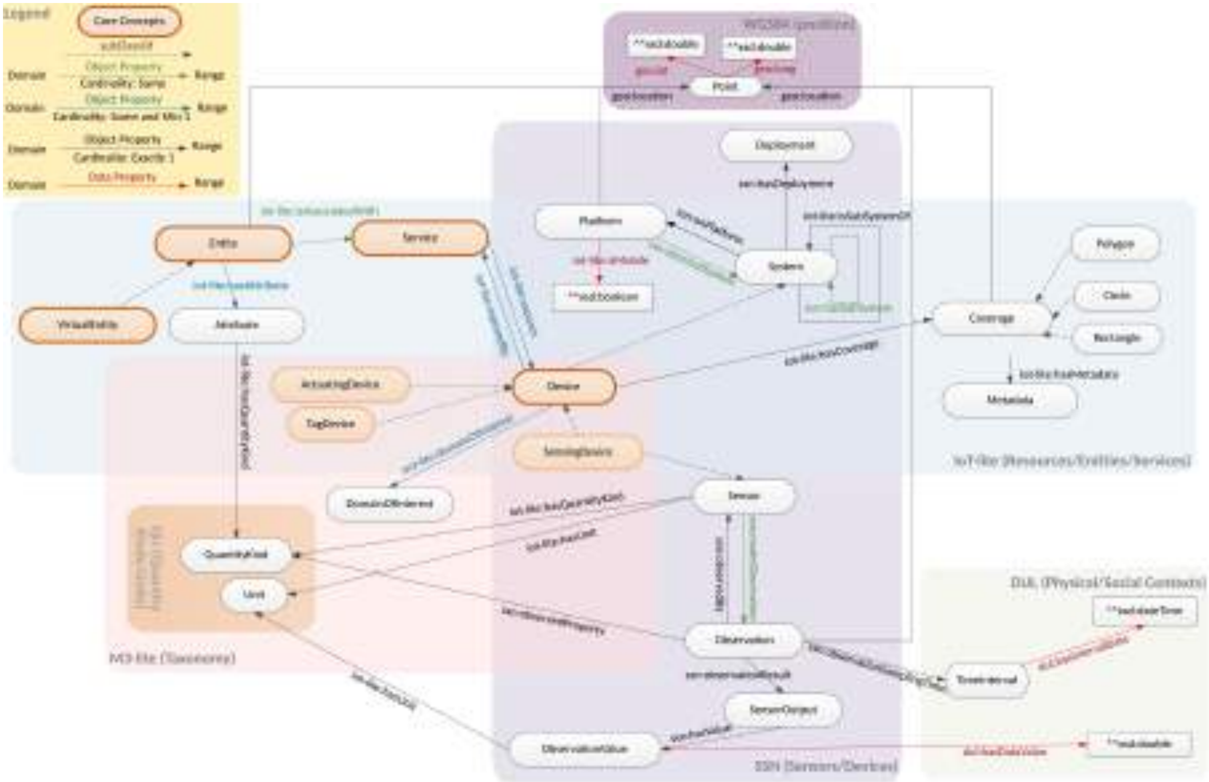


Figure 27: FIESTA-IoT Ontology

3.3.3.1 Statistics

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.

3.3.4 M3-lite taxonomy

The M3 ontology, explained in Section 2.2.6, has been reused to design the M3-lite taxonomy. M3-lite is mainly a taxonomy classifying:

- Resource Type (e.g., Thermometer)
- Quantity kind to describe physical phenomenon sensed by the devices (e.g., Temperature).
- Domain of Interest to describe the different IoT applicative domains (e.g., Smart Home).
- Units (e.g., Degree Celsius) associated to the values produced by devices. It is worth mentioning that there exists a tight relationship between these units of measurement and the quantity kind or physical phenomenon that they are actually measuring.

The M3-lite taxonomy is a lighter version of the M3 ontology, tailored to fulfil the FIESTA-IoT main needs and goals. The refactoring of the ontology has been done to clean non-relevant classes and properties. Further, when the ontologies are reliable (e.g., SSN), instead of using the `owl:equivalentClass` property, we reuse directly the concept already designed in reliable ontology as depicted in Figure 28.

The main purpose of the M3-lite is to extend W3C SSN, more precisely, providing a unified taxonomy for:

- `ssn:Device` concept. A device can be an ActuatingDevice, a SensingDevice or a TagDevice as depicted in Figure 29.
    - `ssn:Sensor` itself has its own taxonomy as depicted in Figure 30 via `ssn:SensingDevice`.
- `qu:QuantityKind` as depicted in Figure 31.
- `qu:Unit` as depicted in Figure 32.

Regarding these last two elements, it is worth stressing the addition of all the quantity kinds and units registered in the four in-house Testbeds that were part of the FIESTA-IoT federation's initial line-up.

Providing such taxonomy is an essential step since IoT data comes from heterogeneous Testbeds using different terms for describing a same observation/measurement.

Figure 28: M3-lite overview

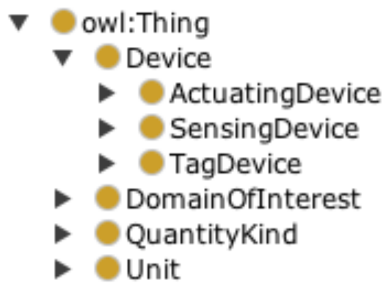


Figure 29: Device Taxonomy

Figure 30: Sensor Taxonomy

Figure 31: A taxonomy for the `qu:QuantityKind` concept

Figure 32 Taxonomy for `qu:Unit` Concept

The M3-lite `DomainOfInterest` concept has been designed to classify 12 popular IoT applicative domains as shown in Figure 33.

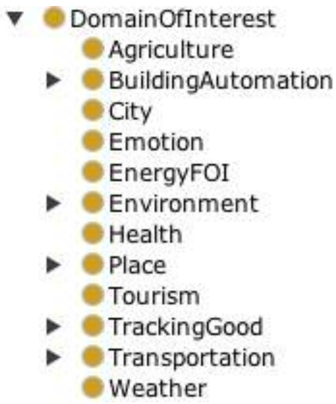


Figure 33: A taxonomy for the `m3-lite:DomainOfInterest` concept

The main benefit of the M3-lite taxonomy is to align and interlink numerous already designed IoT-related ontologies as depicted in Figure 34. Aligning and linking exiting ontologies ease interoperability. The M3-lite taxonomy is even aligned with domain ontologies such as “shw” to describe all sensors used in a smart home and the related knowledge.

“*Ontology networks”*, as explained in the NeOn methodology, would enable the navigation from ontology to another one. We found domain specific ontologies such as music, movie, naturopathy, etc. We decided to integrate them within the M3-lite taxonomy in case we are dealing with RFID tags as well. Hence, by having RFID tags on DVDs or foods and thanks to such domain ontologies, suggesting recipes or home remedies according to the food available in your kitchen is possible or even suggesting the movie for the evening, since the available DVDs in the dining room have been automatically recognized and can be linked to movie ontologies and datasets. 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 ontologies”* in order to support different needs.

Prefix	Value
food_smart_product	http://kmi.open.ac.uk/projects/smartproducts/ontologies/food.owl#
home	http://sensormeasurement.appspot.com/home#
iot-lite	http://purl.oclc.org/NET/UNIS/fiware/iot-lite#
m3-lite	http://purl.org/iot/vocab/m3-lite#
mo	http://purl.org/ontology/mo/
movie	http://www.movieontology.org/2010/01/movieontology.owl
muo	http://purl.oclc.org/NET/muo/muo#
naturopathy	http://sensormeasurement.appspot.com/naturopathy#
ontoSensor	http://mmisw.org/ont/univmemphis/sensor
openIoT	http://openiot.eu/ontology/ns/
owl	http://www.w3.org/2002/07/owl#
qu	http://purl.org/NET/ssnx/qu/qu#
qu_rec20	http://purl.org/NET/ssnx/qu/qu-rec20#
qudt	http://qudt.org/schema/qudt#
qudt_unit	http://data.qudt.org/qudt/owl/1.0.0/unit.owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
shw	http://paul.staroch.name/thesis/SmartHomeWeather.owl#
spitfire	http://spitfire-project.eu/ontology/ns/
ssn	http://purl.oclc.org/NET/ssnx/ssn#
sweet_unit	http://sweet.jpl.nasa.gov/ontology/units.owl#
txn	http://lod.taxonconcept.org/ontology/txn.owl#
ucum	http://idi.fundacionctic.org/muo/ucum-instances.owl

Figure 34: The M3-lite taxonomy reusing existing IoT ontologies

3.3.5 FIESTA-IoT ontology and M3-lite Taxonomy Documentation

To encourage the easy reuse of the ontologies, we employed several documentation tools:

- **Protégé**[39], a popular ontology editor is used to generate the OWL version the FIESTA-IoT ontology and the M3-lite ontology. The Figure 28 - Figure 32 above show the M3-lite taxonomy as a tree in Protégé. Such visualization is relevant when designing the ontology. Further, a plugin called OWLDocs is available that can be used to generate the documentation of the FIESTA-IoT ontology and the M3-lite taxonomy in the HTML version.
- **Parrot**[40] is an online tool easy to use [61] and does not require setting up any code on the local machine to generate the documentation in the HTML version.
- **WebVOWL**[41] is a visualization tool that is easy to use. It requires the namespace of the ontology and automatically displays a graph-based visualization. Figure 35 and Figure 36 show the visualizations for the M3-lite taxonomy using WebVOWL[42].
- **LODE [43] (Live OWL Documentation Environment)** automatically generates a documentation to assist people in employing the taxonomy [62]. We currently use it to generate the HTML version of the M3-lite taxonomy. Figure 37 and Figure 38 show the HTML version of the FIESTA-IoT ontology and M3-lite Taxonomy dynamically generated by LODE configured specially for FIESTA-IoT project. We wish to automate the process by running the version of the LODE tool on FIESTA-IoT platform so that it includes the automatic update feature of the web pages whenever the new version of the ontology is released.

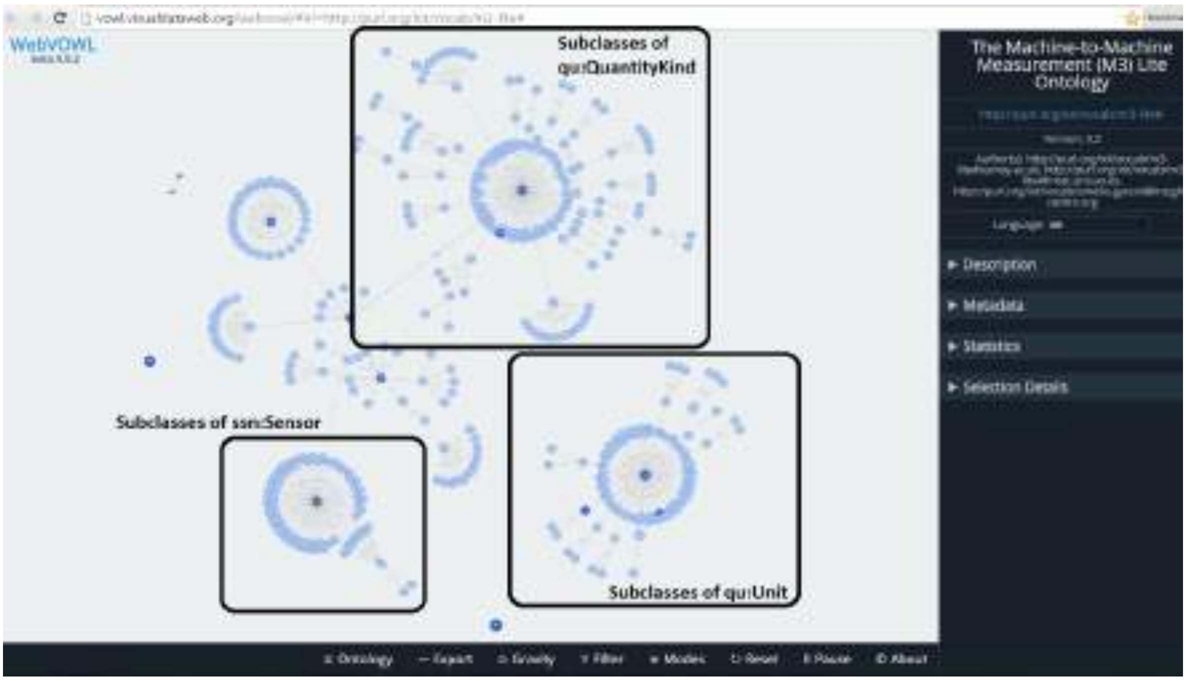


Figure 35: M3-lite Taxonomy visualized with WebVOWL

[39] http://protege.stanford.edu/
[40] http://ontorule-project.eu/parrot/parrot
[41] http://vowl.visualdataweb.org/webvowl/
[42] http://vowl.visualdataweb.org/webvowl/#iri=http://purl.org/iot/vocab/m3-lite#
[43] http://www.essepuntato.it/lode

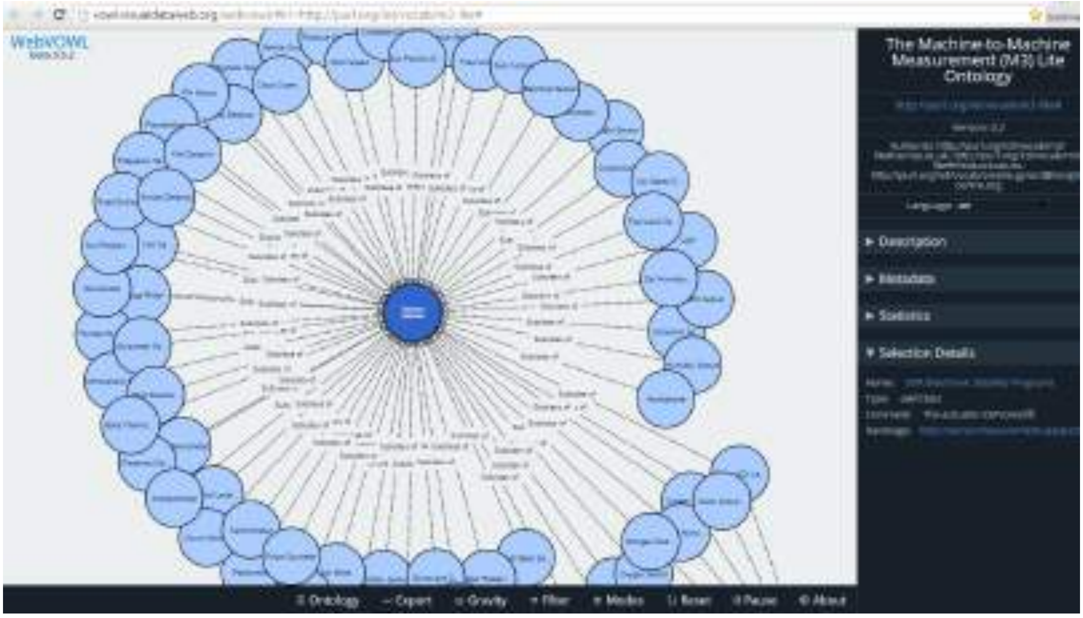


Figure 36: `ssn:sensor` designed within M3-lite Taxonomy visualization using WebVOWL

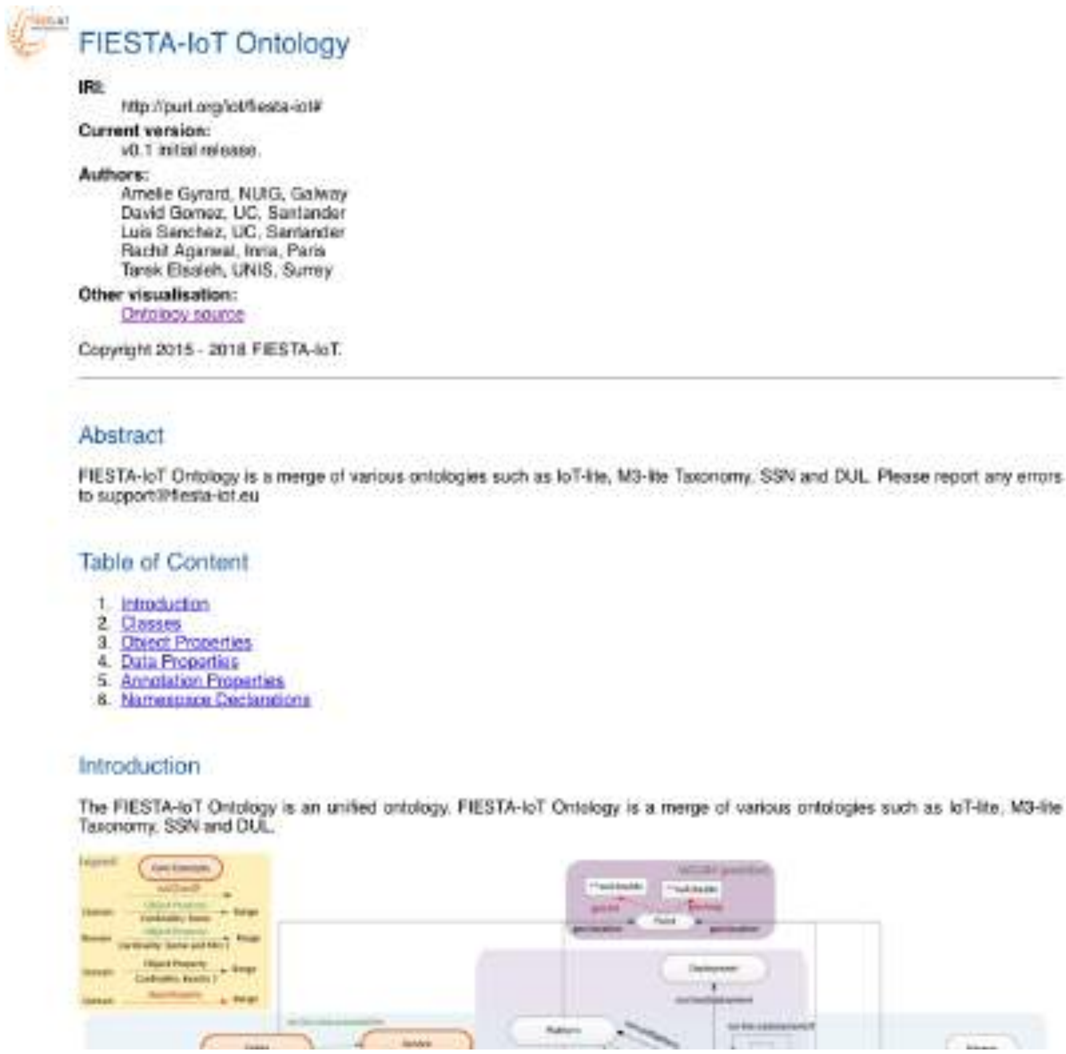
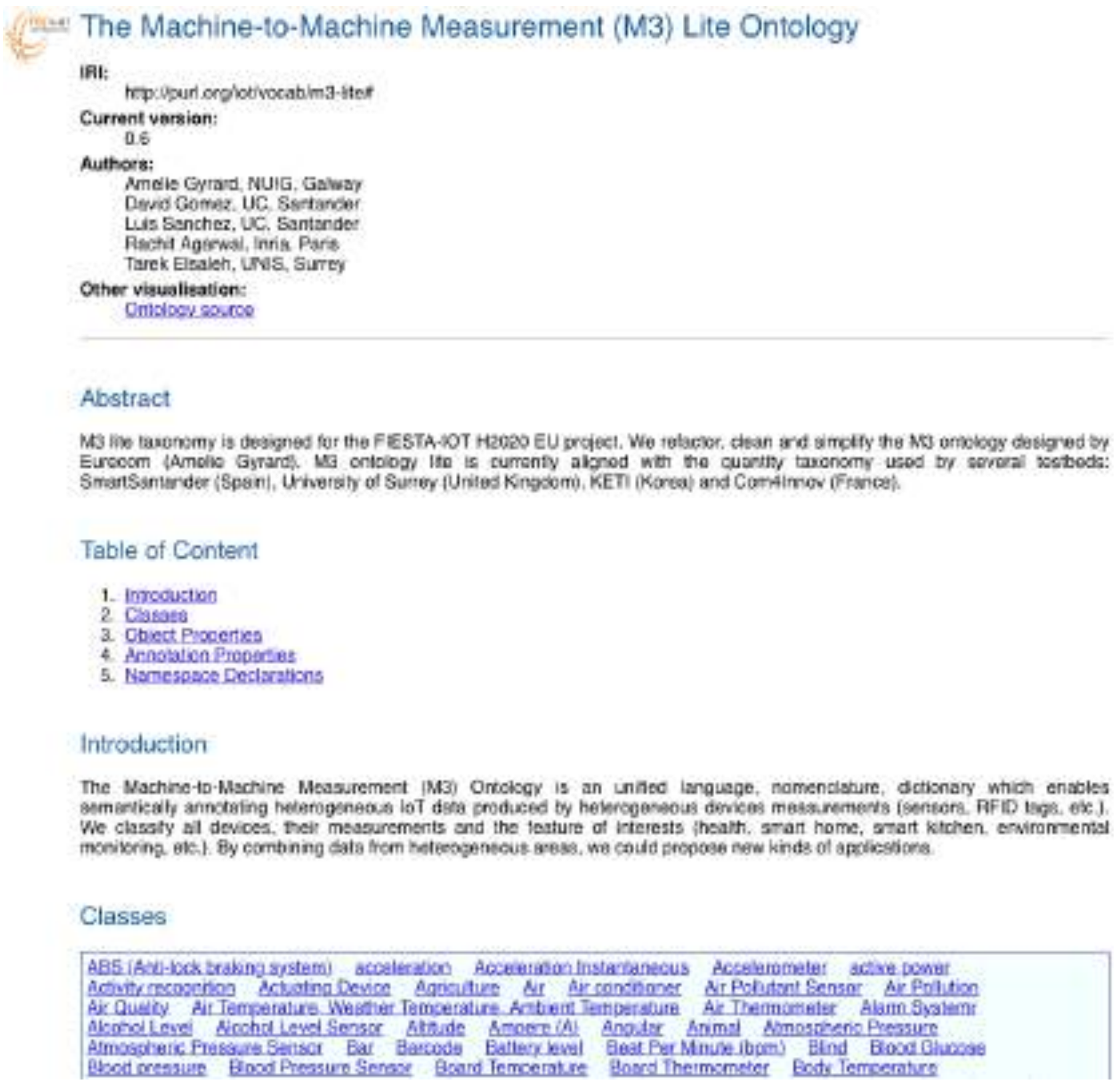


Figure 37: FIESTA-IoT ontology document generated with LODE



The Machine-to-Machine Measurement (M3) Lite Ontology

IRI:
http://purl.org/iot/vocab/m3-lite#
Current version:
0.6
Authors:
Amelie Gyrard, NUIG, Galway
David Gomez, UC, Santander
Luis Sanchez, UC, Santander
Rachit Agarwal, Inria, Paris
Tarek Elsaleh, UNIS, Surrey
Other visualisation:
Ontology source

Abstract

M3 lite taxonomy is designed for the FIESTA-IOT H2020 EU project. We refactor, clean and simplify the M3 ontology designed by Eurecom (Amelie Gyrard). M3 ontology lite is currently aligned with the quantity taxonomy used by several testbeds: SmartSantander (Spain), University of Surrey (United Kingdom), KETI (Korea) and Com4Innov (France).

Table of Content

1. Introduction
2. Classes
3. Object Properties
4. Annotation Properties
5. Namespace Declarations

Introduction

The Machine-to-Machine Measurement (M3) Ontology is an unified language, nomenclature, dictionary which enables semantically annotating heterogeneous IoT data produced by heterogeneous devices measurements (sensors, RFID tags, etc.). We classify all devices, their measurements and the feature of interests (health, smart home, smart kitchen, environmental monitoring, etc.). By combining data from heterogeneous areas, we could propose new kinds of applications.

Classes

ABS (Anti-lock braking system) acceleration Acceleration Instantaneous Accelerometer active power
Activity recognition Actuating Device Agriculture Air Air conditioner Air Pollutant Sensor Air Pollution
Air Quality Air Temperature, Weather Temperature, Ambient Temperature Air Thermometer Alarm System
Alcohol Level Alcohol Level Sensor Altitude Ampere (A) Angular Animal Atmospheric Pressure
Atmospheric Pressure Sensor Bar Barcode Battery level Beat Per Minute (bpm) Blind Blood Glucose
Blood pressure Blood Pressure Sensor Board Temperature Board Thermometer Body Temperature

Figure 38: M3-lite Taxonomy document generated with LODE

3.3.6 SPARQL Queries on the FIESTA-IoT ontology

In this section, we provide few examples of SPARQL queries that can be run on the data that satisfies the FIESTA-IoT Ontology.

- The first sample query is the query to get all information about the resources of type `ssn:SensingDevice`

```
PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
SELECT *
WHERE {
  ?resource a ssn:SensingDevice.
}order by asc(UCASE(str(?s)))
```

- The second sample query is the query to get resources that are of type `ssn:SensingDevice` and are based in certain area characterized by their geographical location.

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
PREFIX ssn: <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
```

```
SELECT ?s ?lat ?lng ?endp
    WHERE {
        ?s a ssn:SensingDevice .
        ?s iot-lite:isExposedBy ?serv .
        ?s iot-lite:hasQuantityKind m3-lite:AirTemperature .
        ?s ssn:onPlatform ?platform .
        ?platform geo:location ?point .
        ?point geo:lat ?lat .
        ?point geo:long ?lng .
        ?serv iot-lite:endpoint ?endp.
        FILTER (?lng > -50 && ?lng < 50 && ?lat < 100 && ?lat > 0 )
    }
```

4 TOOLS

In this section we provide the overview of FIESTA-IoT Annotation tools and Validation tools that uses FIESTA-IoT ontology developed in Section 3.3.3.

4.1 FIESTA-IoT Annotation Tool

For IoT Testbeds joining the FIESTA-IoT platform, the descriptions that correspond to their resources and data must conform to a common model, i.e. the FIESTA-IoT ontology as described in Section 3.3.3. From Task 2.4 [1], Testbeds have been distinguished in three classes. Class-I Testbeds corresponds to a Testbed that has a semantic persistence for the resource and data descriptions that already conforms to the FIESTA-IoT ontology (i.e., 100% compliant). Pertaining to Class-II and Class-III Testbeds, they do not internally “speak” the FIESTA-IoT annotated format (i.e. following the FIESTA-IoT ontology) and thus, they must rely on an annotator that “translates” from their intrinsic format (e.g. FIWARE-compliant, OneM2M, raw JSON, etc.) to the one that is understood and interpreted within the FIESTA-IoT platform.

With regards to these annotation tools, it is worth highlighting that they will not run as part of the FIESTA-IoT platform, but will be executed at Testbed level. Therefore, the different Testbed providers that are willing to become part of the FIESTA-IoT federation must deal with the implementation and management of their own semantic annotators, thus making their own IoT services/resources/data/VE descriptions compatible with the ontology that has been defined throughout this deliverable. However, all the annotators developed by the FIESTA-IoT first-parties will be completely available for external users so that they can tailor their corresponding formats to that of FIESTA-IoT, without the need of starting their own annotator from scratch, but they can leverage the guidelines brought about by the acquaintanceship and experience acquired by other Testbeds that have already developed their own solutions. Namely, from our initial batch of four in-house Testbeds, each of them showcase a number of particularities that aim to span a high range of possibilities so that third parties might find inspiration to define their own annotators. To have a deeper and more detailed information about this, the reader should refer to [63] and Task 3.2 deliverable, where a verbose description of each of these native formats are addressed.

- SmartSantander follows a proprietary format based on the utilization of JSON schemas.
- UNIS, like SmartSantander, uses a proprietary non-standardized format which represents data in either XML or JSON format. The data description represents an “observation” from the Smart ICS IoT node, which is essentially a set of readings from a device.
- Com4Innov is a FIWARE-compliant Testbed. As such, it relies on the NGSI 9/10 interfaces to describe resources and observations, respectively.
- KETI bases it annotation on the oneM2M standard.

Sticking to the annotation of one (arbitrarily chosen) of the SmartSantander's resources/devices, we show in Figure 39 and Figure 40 an illustrative example of how this concrete annotator works, focusing on the description of a single Device (i.e. urn:x-iot:smartsantander:u7jcfa:t10000) and one of the measurements generated by one of its sensors.

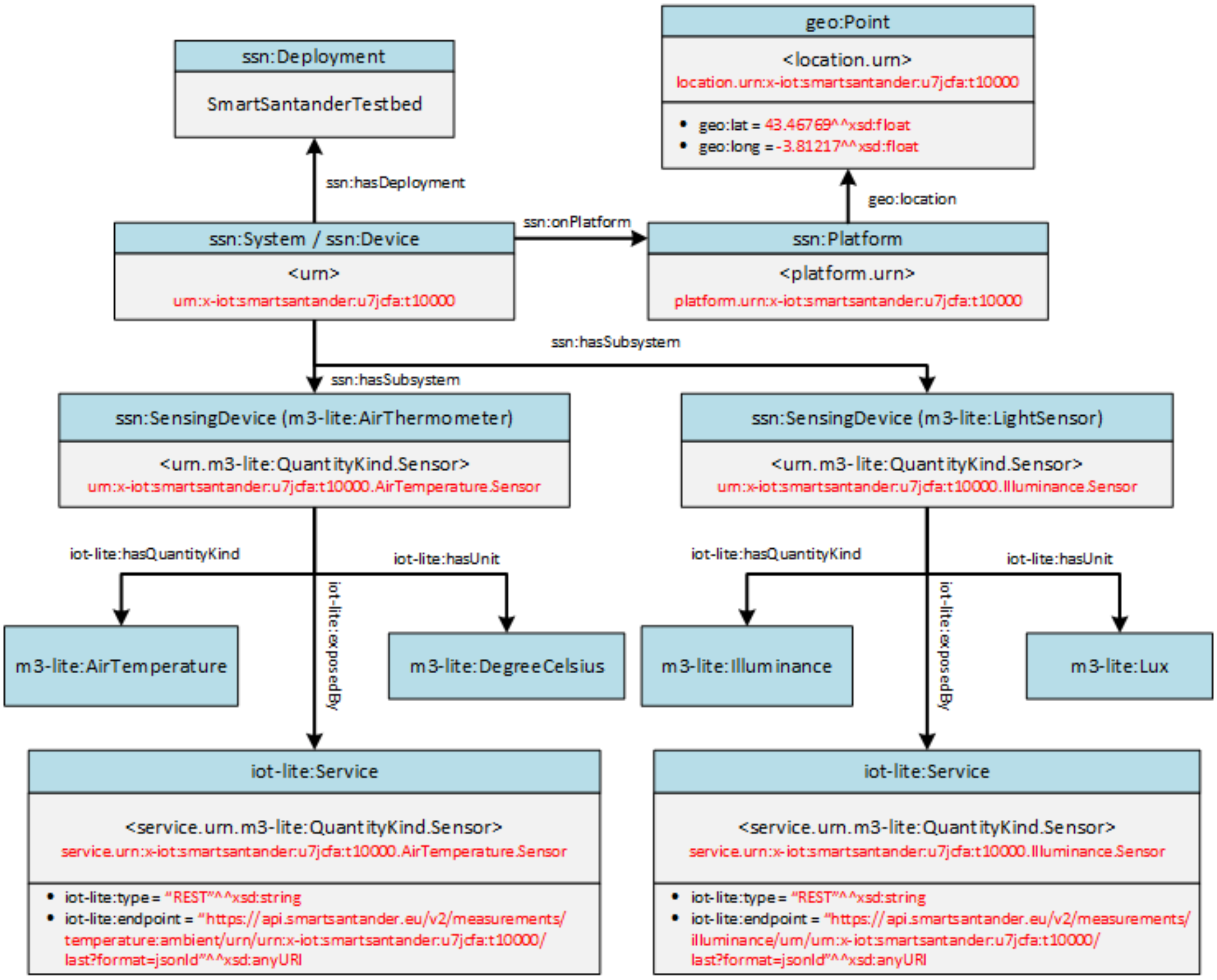


Figure 39: SmartSantander annotator example (Graph of a resource description)

One of the first steps to be carried out when a Class II or Class III Testbed becomes part of the FIESTA-IoT ecosystem[44], much before the gathering of data from the devices, consists in the registration of all its resources and IoT services. In Figure 39 we can observe the template designed to map from the legacy SmartSantander format to the one elicited for fulfilling the FIESTA-IoT ontology, commented in Section 3.3.3. As commented there, the root of the graph (i.e. `ssn:Deployment`) refers to the actual Testbed or platform which the corresponding resource belongs to. Below this we can find the identification of the `ssn:Device` itself, bound to a physical location (which is hooked via `ssn:Platform`). As asserted before, one single device might host a number of `ssn:SensingDevices`, just like we can see in this example, where we do observe an ElectricalSensor (that measures the battery level remaining), an Air Thermometer (that measures the ambient temperature) and, at last, a Light Sensor that gathers the information about the illuminance perceived in the ambient. Please note that the naming of these elements match the ones defined in the M3-

[44] The operation for Class I Testbed does not require the registration of resources. Moreover, in this first stage in the project's lifetime, the SmartSantander platform matches to the definition of a Class II Testbed.

lite taxonomy, namely the ones that correspond to the Resource Type class. As it could be easily inferred, SensingDevices are tied to a pair of QuantityKind/Unit, which makes an univoque bound between the resource type and the physical phenomenon they are collecting information of. Finally, we need to annotate the IoT Service that will expose each of the resource, i.e. to retrieve the last value measured. It is worth highlighting that these services will be linked to the SensingDevices, hence every measurement/observation will only contain information of one physical phenomenon.

After describing the resource annotation format, one of the things that is worth bringing up again is, as hinted in Section 3.3.3, the approximation to the concept of Web of Things in those entities that could brought about useful information to be consumed using standard web protocols. Commented above, we have limited this feature to those entities that can potentially expose a service, i.e. `ssn:Device` and `ssn:SensingDevice`. Then, these elements could be directly accessed through e.g. a straightforward HTTP request, thus experimenters might browse among the different resources that compose the whole just like they would surf the web, by clicking in those links they are interested in. Seen from another perspective, we are giving FIESTA-IoT's things their own public URLs, so they might be indexed by popular web engines or, even a step beyond the traditional paradigm, they might be used in services like IFTTT[45] (IF This Then That) or Node-RED[46], where users can create their own applications by combining the information of our devices with their own self-created rules to create a new level of knowledge.

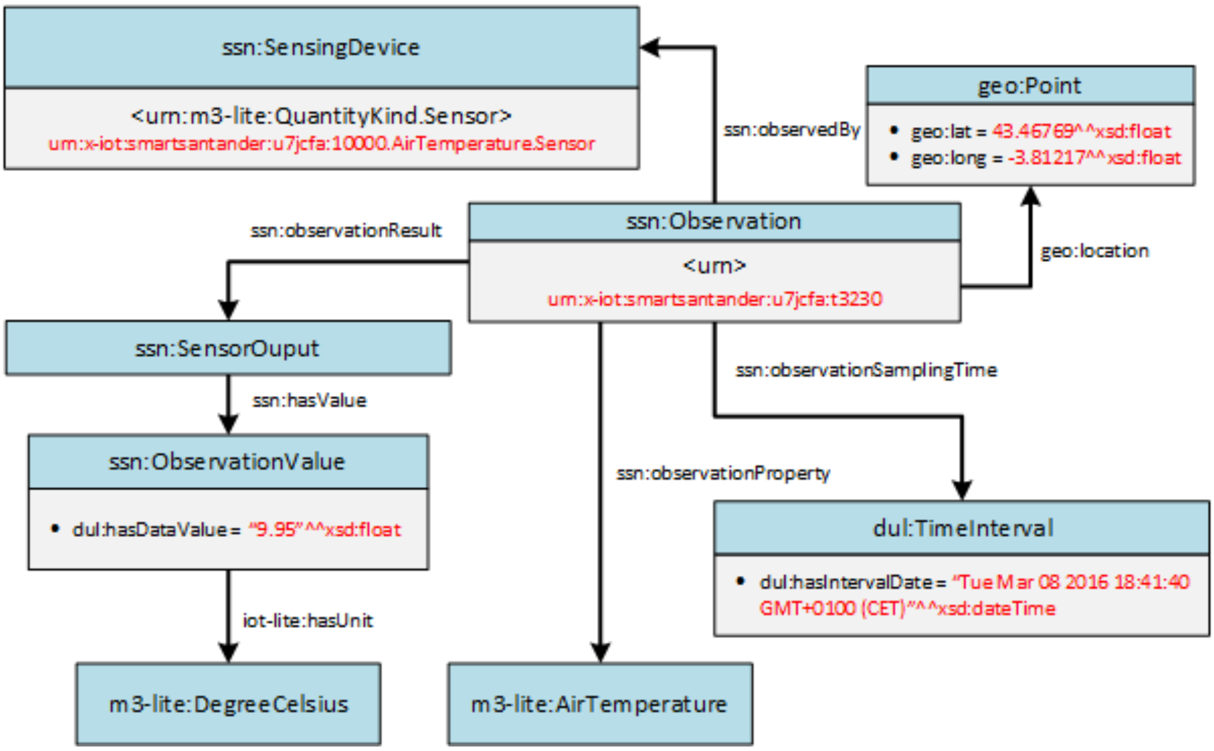


Figure 40: SmartSantander annotator example (Graph of an observation)

Now focusing on the annotation of the data, Figure 40 displays the resulting graph of one (again arbitrarily selected) of the measurements generated by the above-mentioned resource. The first and foremost element to be included is the identity of the SensingDevice that carried out the observation per se (recall that the IoT services focus on the `ssn:SensingDevice` class and not its immediate upper level (`ssn:Device`)). Below, the `ssn:Observation` concentrates all the parameters that shape the observation, split, as mentioned before, into five main aspects:

[45] http://ifttt.com
[46] http://nodered.org/

- Physical position of the SensingDevice (right side). We must remark here that, thanks to the inheritance properties of `ssn:SensingDevice` with respect to its base class, `ssn:Device`, every sensor installed in a Device will share its GPS coordinates, mapped through the `geo:lat` and `geo:long` tuple.
- Timestamp: Apart from the value measured by the sensor, the time is most important to include, as part of the measurement, in which the observation was taken, following a wide variety of formats supported by the data type belonging to the `dul:TimeInterval` (e.g. `xsd:gYear`, `xsd:dateTime`, etc.). Thanks to this timing notation, more complex IoT services, based on historical searches, might be possible.
- Quantity Kind. Said in few words, each sensor (or type of sensor) must be intimately linked to a physical phenomenon. As commented in Section 3.3, the FIESTA-IoT ontology relies on the M3-lite taxonomy to describe and annotate the phenomena retrieved by physical resources.
- Unit of measurement. Although the relationship between a SensingDevice and a Quantity Kind is “atomic” and shall not be modified, it is also true that units might be prone to change, since sensors might alter their configuration at some moment in time, e.g. to return meters per second instead of kilometers per hour. Even, sensors measuring the same physical phenomenon (i.e. different manufacturers) might rely on different units of measurement. After all, every value generated by a measurement will be tied to the unit reflected in this object property.
- Value of the measurement. Undoubtedly, the cornerstone of the document/graph. The actual value “observed” by the physical sensor. As can be easily inferred, different data types might be applied here (e.g. `xsd:integer`, `xsd:float`, etc.).

As has been already highlighted, this is one illustrative example of how both SmartSantander’s resources and measurements descriptions are annotated according to the formats defined in the FIESTA-IoT ontology (see Section 3.3.3).

4.2 FIESTA-IoT Validation Tool

An important requirement in order to maintain interoperability of information originating from different Testbeds is to make sure they comply with the common model mandated by the FIESTA-IoT platform. Therefore a validation mechanism is required to check resource and data descriptions compliance. Several approaches can be taken to enable this:

- The first is to provide a tool that Testbed providers can use to test their annotations against the FIESTA-IoT model. This involves a proactive offline process conducted by a Testbed provider before registering with the FIESTA-IoT federation.
- The second would be to perform validation upon the new registration of a Testbed and its Resources with the FIESTA-IoT platform. Here, the Resource Manager will forward the Resource(s) that is to be registered to the validation mechanism process.
- The third involves an auditing process whereby the annotated data that is published onto the FIESTA-IoT platform is occasionally checked for compliance.

During the validation process, the validator can perform one or more types of checks. The checks are either Syntactic or Semantic. Syntactical inconsistences within a description can include:

- Unknown properties and classes with respect to the FIESTA-IoT model,
- Problematic prefix namespaces,
- Ill-formed URIs and language tags on literals,
- Data-typed literals with illegal lexical forms,
- Unexpected local names in schema namespaces,
- Untyped resources and literals,
- Individuals having inconsistent types, assuming complete typing broken RDF list structures.

While semantic inconsistencies within a description can include:

- Inheritance relationships for classes and properties,
- Cardinality,
- Expected domains and ranges.

In FIESTA-IoT, the SSN validator (see Section 2.4.1) is being adopted as a base asset, and will be modified to include all models that form the FIESTA-IoT ontology. The SSN Validation Service mainly checks for syntactic issues, whereas semantic issues are not covered. Basic semantic validation will be applied, and evaluated to see if the process does not introduce significant delays.

5 FIESTA-IOT MOBILITY MANAGEMENT

Mobility characteristics, properties and patterns are present in all mobile devices and systems. Testbeds have mobile devices that move from one place to another to capture the required phenomenon (QuantityKinds). In FIESTA-IoT, currently the Testbed that have mobile devices are SmartSantander Testbed and Com4Innov Testbed. More Information about the Testbeds associated with FIESTA-IoT platform can be found in [63]. However, note that as a part of the open call and in future, more Testbeds having mobile devices can join FIESTA-IoT.

5.1 Analysis for the need of dedicated Mobility Management Component

The need for the mobility management component first leads us to answer the following related questions:

- How to handle Inter-gateway mobility?
  Here, the challenge is how to ensure that a mobile device (e.g., using Bluetooth) stays connected while moving within its Testbed. Note that as FIESTA-IoT platform scope is federation between Testbeds from where FIESTA-IoT platform is currently getting data (resource description, actual data produced) from Testbed, it is assumed that the Testbed will handle mobility management aspect such as handover.
- How to handle Inter-Testbed mobility?
  The use case here would be as follows: A Testbed “A” for sensing ambient noise is implemented in a city “A” and uses mobile devices, whereas a Testbed “B” which also senses ambient noise is also implemented but in city “B” and also uses mobile devices. These Testbeds have been developed independently, and store data in their separate clouds; made available to FIESTA-IoT experimenters through the FIESTA-IoT meta-cloud. The data produced by the devices in both the Testbeds have associated metadata that consists of location information.
  When a device from Testbed “A” moves to the location that is served by devices in Testbed “B”, can the device from Testbed “A” be associated to Testbed “B”?
  This would need an agreement between the Testbeds so that devices can be associated to either Testbed depending on their current location. This feature is not in the scope of FIESTA-IoT i.e. FIESTA-IoT does not provide/perform agreements between Testbeds. Thus this question is also irrelevant with respect to FIESTA-IoT. FIESTA-IoT provides federation of the Resource descriptions and IoT data (measurements with metadata). FIESTA-IoT does not control the Testbeds. Thus the “inter” Testbed device sharing is not in the scope of FIESTA-IoT.
- How to handle multiple Testbeds sharing the same physical space?
  Building upon the above, this brings back the idea from MobIoT ("I want current values of road traffic speed and ambient noise from Paris") with the special case that

the devices are not owned/managed by the same Testbed. The use case here is that there are two Testbeds available within same physical space with each providing access to their data through FIESTA-IoT. FIESTA-IoT does not manage Testbed, thus as said before at the physical level Testbeds should have agreement among each other. As one of the federation mechanisms, FIESTA-IoT federates data provided by the devices in various Testbed associated with FIESTA-IoT platform.

Now with respect to the data following aspects must be addressed:

- Data provided from Testbed to FIESTA-IoT: As devices are mobile, the Resource descriptions and IoT data produced is frequently updated and sent to FIESTA-IoT platform. Thereby requiring updates to Resource Registry and subscription Manager whenever a device is moved. To address this:
  - As Testbed knows when the device has moved, the resource description of a resource in FIESTA-IoT platform should be updated by the Testbed. However this involves excessive network overload, connection problems, etc. as devices would move too frequently. To overcome this issue as Measurements are accompanied with spatio-temporal information, FIESTA-IoT data manager component would be one of the options to update the resource registry and subscriptions. Nevertheless, this approach has issues like: resource description (having metadata like location) being old, need a mechanism to run the component within data manager to push for updates as soon as data is available. Further, another approach to overcome this issue is to have a real time messaging paradigm - publish-subscribe - where the Testbeds can update the FIESTA-IoT resource registry whenever the Testbed has an updated resource description. Such publish-subscribe requirement can be handled via the implementation of `RabbitMQ`[47] client and server. However, this requires: (a) Testbed (publisher) to run an additional FIESTA-IoT publishing component that would publish to a Testbed specific queue on the Message Broker provided by the `RabbitMQ` server, (b) FIESTA-IoT Resource Manager (subscriber to events published) to run `RabbitMQ` subscription component that would pull from the Message Broker provided by the `RabbitMQ` server. Note here the queue size would be chosen based on number of devices in the Testbed and the Time to Live (TTL) for the message can be handled via freshness of the message.
- Data provided from FIESTA-IoT platform to the experimenters: For the case, when two or more devices with same properties are associated to a same location (identified by latitude and longitude) one issue that must be addressed is that which data from which Testbed should be provided. The solution to this problem is:
  - If the request is made to the resource registry then each device is different. Thus all the device resources available in that location will be provided.
  - If measurement is requested then it depends on the FIESTA-IoT policies of which data should be provided. FIESTA-IoT could provide
    - The average of the sensed values for the location, or
    - The latest sensed value for the location.

  Note that this case is only applicable when two devices within FIESTA-IoT space share the same location. When the devices have different locations, the FIESTA-IoT platform returns experimenters the needed data.

Thus, from the above analysis it can be deduced that there is no need for FIESTA-IoT platform to handle device mobility: however, FIESTA-IoT platform should handle updates to the resource registry within the FIESTA-IoT platform. The update to the resource registry can be made either by: (a) Data management component that perform the update to the resource registry and subscriptions (b) using publish-subscribe approach using `RabbitMQ`.

[47] https://www.rabbitmq.com

5.2 Mobility Related Ontology

Basic mobility support in the ontology is provided. All observations have location and timestamp associated. Mobility of a device can be inferred from such location and timestamp attributes in the measurement. However, Devices are attached to platform. Thus, it is imperative to know whether the platform is mobile or not. To facilitate the knowledge whether the platform is mobile or not, "`isMobile`" attribute can be made available in the ontology. This attribute is used in the resource description.

Another aspect related to mobility is storing the inferred mobility patterns and properties semantically from the data that are provided by the Testbeds. This requires inference engine to be available in order to analyse the patterns and properties and then store them semantically. For this data there are no well-defined ontology available that stores mobility patterns and properties. Rather trajectory ontology such as that of Athena unified trajectory ontology [64] and [65], [66] is available. The Figure 41 shows the Athena trajectory ontology. Note that mobility patterns and properties are wide, some of which are speed, direction of motion, regular trip, accompanying details, personal places, periodic nature etc. A brief about the mobility patterns and properties is available at [51]. In future we would like to investigate how we can update the current FIESTA-IoT ontology to include the mobility properties and patterns with:

- (Must) speed and direction of motion
- (Could) mobility related properties, patterns and characteristics (such as regular trips, personal places etc.).

Furthermore, note that no in-house Testbeds currently produces inferences on the mobility data. Also, in FIESTA-IoT, we do not intend to do data analytics to generate mobility patterns and properties. As a knowledge producer, if a Testbed or an Experiment wants to generate or has mobility patterns and properties data stored and wish to provide such information to FIESTA-IoT, inclusion of properties with respect to mobility patterns and properties can be useful.

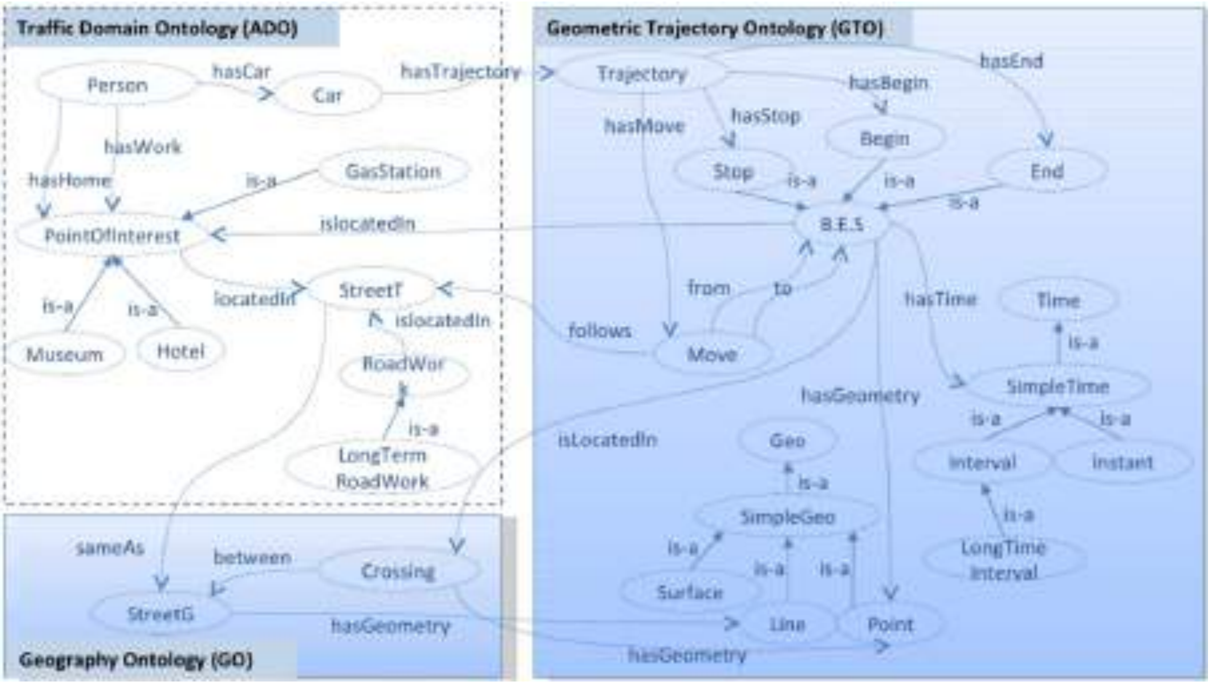


Figure 41: Athena Trajectory Ontology

6 GUIDELINES FOR TESTBEDS AND BEST PRACTICES TO PUBLISH IOT DATA IN FIESTA-IOT

When it comes to publishing descriptions about FIESTA-IoT Resources and their corresponding data, best practises should be applied to enhance efficiency in the FIESTA-IoT framework. Annotated data provide benefits by providing context about the readings or observations that are captured from the real world. But at the same time 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 can be used as a guide to produce better data:

- Registration of Resources 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.
- Derefenceable URIs must only be applied to instantiations of FIESTA-IoT Resources, which in the ontology refers to the “`ssn:Device`” class or any of its subclasses (`ssn:SensingDevice`, `iot-lite:ActuatingDevice`, `iot-lite:TagDevice`).
- Descriptions can be annotated using any RDF serialization format, which include RDF/XML, JSON-LD, Turtle, or N3.
- It is expected to handle large amounts of data being produced by the Testbeds, and so annotation applied to raw/proprietarily-formatted data should be minimal, and any triples created must provide added-value for experimenter queries. This will alleviate unnecessary load on the platform or Testbed when it comes to delivery and storage.
- Prior to publication, Testbed providers are encouraged to validate samples of their descriptions with the FIESTA-IoT ontology validator.

7 CONCLUSION AND FUTURE WORK

This deliverable focuses on how data available from different Testbeds can be stored and utilized within the FIESTA-IoT framework, including important aspects like interoperability and federation. In order to do so, semantic approach has been adopted. Each Testbed, in order to be the part of the FIESTA-IoT federation, will have to comply with the defined FIESTA-IoT Ontology and will have to annotate the data they are providing to FIESTA-IoT. This annotation will be validated, stored and finally provided to the experimenters upon their request.

To achieve the above flow, we have surveyed the state of the art ontologies with respect to the available semantic models in the literature. The motivation to build the FIESTA-IoT 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 fulfil the needs of the Testbeds, (ii) reusing as much possible the existing ontologies in order to help Testbeds not re-annotate their datasets in order to join FIESTA-IoT Federation, and (iii) ensuring a better interoperability with existing semantics-based IoT platforms, projects, and standardizations. The state of the art ontologies discussed in the deliverable also provide essence on why the ontology is suitable for being merged within FIESTA-IoT ontology.

Thus, based on the initial analysis, the initial version of FIESTA-IoT ontology currently relies on concepts from IoT-lite, SSN, QU, WGS84 and DUL ontologies and M3-lite taxonomy. IoT-lite integration helps “In-house” Testbeds, such as that of UNIS Testbed to not to recreate annotations that they already have (although minor changes would be required). SSN, probably the most well-known one pertaining to the sensing realm, provides the basic concepts, while M3-lite provides, above all, the taxonomy that cover all the different types of resources, physical phenomena and units of measurement fostered from the underlying

Testbeds (it is worth highlighting that this taxonomy will be a living entity that might be updated with all the new elements coming from potential new Testbeds that become part of the FIESTA-IoT federation). Furthermore, we are planning to include concepts from other ontologies, like SAO, OpenIoT and VoID, and other standardizations like oneM2M (which will have a big impact in the future). Above all, best practices have been followed within the ontology definition process. Within this frame; the ontology documentation, some sample annotations and the mapping between legacy Testbeds' and M3-lite's QKs and units (this last part, done by In-house Testbeds) is also provided. Thus, external Testbeds could use this as a guideline to annotate their resources (and future observations) to the format defined in the scope of the FIESTA-IoT federation.

Moreover, we have also described some of the off-the-shelf annotation tools. Nevertheless, as the data is provided by the Testbeds, it is worth noting that FIESTA-IoT platform does not execute or provide such annotation tools. FIESTA-IoT only provides a reference annotation tool that can be used by the Testbeds as a base for the implementation of their own annotators. Apart from this annotation process, it is deemed necessary to validate the data injected from the underlying Testbeds in order to corroborate that the different datasets accomplish the templates defined in the FIESTA-IoT ontology. This validation will only happen based on the policies (when and what to validate) specified by FIESTA-IoT, and will be carried out within the FIESTA-IoT platform.

As Testbeds might provide IoT data coming from mobile devices, it is worth investigating issues related to how this data will be stored and how the updates will be published to FIESTA-IoT platform. The need for real-time updates would be satisfied by the publish-subscribe methodology and a component developed by FIESTA-IoT could be integrated by Testbeds within their framework.

In addition, this deliverable also provides insights to the guidelines and best practices for Testbeds to publish data (i.e. measurements) through the FIESTA-IoT platform, either to store it or to forward it to experimenters. Such guidelines would ensure less overhead for FIESTA-IoT in terms of data management and its usage.

However, there are number of open issues that have not been included in this version of the deliverable and will be addressed in the future iteration. Below we include a list of such currently open issues:

- The way to store the different datasets (i.e. IoT services/resources, measurements, virtual entities, Testbeds, rules, etc.) is a point that will be thoroughly analyzed in this v2. The most likely solution we have foreseen is to make use of JENA-based triple stores instanced in various functional components throughout the platform (following the above example, IoT Service/Resource Directory, Meta-Cloud Data Endpoint, Virtual Entity Registry, the Built-in Reasoner, etc.).
- Once we have saved the various data in the platform, it will have to provide a way to interact with so that experimenters/knowledge producers/value-added service providers can extract the information according to their needs. For that, the most widespread solution is no other but SPARQL.
- As it was mentioned before, neither the FIESTA-IoT ontology nor the M3-lite taxonomy are finished versions. As long as we discover missing elements (e.g. coming from the feedback from other tasks or external Testbeds fostered from the Open Call processes) or improvable nodes, we will modify and update the ontology, which was originally presented in Figure 27. Some of the aspects along these lines include: how stream data can be incorporated, how mobility related concepts can be added in the ontology, what taxonomy to use for composed/inferred IoT data, how to align similar concepts that have same characteristics in different ontologies.

8 REFERENCES

[1] FIESTA-IoT, “Deliverable 2.4: FIESTA-IoT Meta Cloud Architecture,” 2015.

[2] IoT-A, “Deliverable D1.5: Final Architectural Reference Model for the IoT,” 2013.

[3] IoT-A, “Deliverable D1.2: Initial Architectural Reference Model for IoT,” 2011.

[4] IEEE, “Guide for Monitoring, Information Exchange, and Control of Distributed Resources Interconnected with Electric Power Systems,” 2007. [Online]. Available: http://goo.gl/gEQlqd. [Accessed: 31-Mar-2016].

[5] V. Haren, “TOGAF Verson 9.0,” 2009.

[6] A. Gavras, “Experimentally Driven Research White Paper,” 2010.

[7] IEEE, “IEEE Standard Glossary of Software Engineering Terminology,” 1990. [Online]. Available: http://goo.gl/svqXqB. [Accessed: 31-Mar-2016].

[8] A. H. Soukhanov, K. Ellis, and M. Severynse, *The American Heritage Dictionary of the English Language*. Boston: Houghton Mifflin, 1992.

[9] A. Gyrard, M. Serrano, and G. A. Atemezing, “Semantic web methodologies, best practices and ontology engineering applied to Internet of Things,” in *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 412–417.

[10] J. Ye, L. Coyle, S. Dobson, and P. Nixon, “Ontology-based Models in Pervasive Computing Systems,” *Knowl. Eng. Rev.*, vol. 22, no. 04, pp. 315–347, Dec. 2007.

[11] M. Compton, C. Henson, L. Lefort, H. Neuhaus, and A. Sheth, “A Survey of the Semantic Specification of Sensors,” in *Proc. of 2nd International workshop of Semantic Sensor Networks*, 2009, pp. 17–32.

[12] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, “The SSN ontology of the W3C semantic sensor network incubator group,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.

[13] L. M. Ni, Y. Zhu, J. Ma, Q. Luo, Y. Liu, S. C. Cheung, Q. Yang, M. Li, and M. you Wu, “Semantic Sensor Net: an Extensible Framework,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 4, no. 3/4, pp. 157–167, 2009.

[14] D. Bell, B. R. Heravi, and M. Lycett, “Sensory semantic user interfaces (SenSUI): position paper,” in *2nd International Conference on Semantic Sensor Networks*, 2009, vol. 522, pp. 96–109.

[15] M. Calder, R. A. Morris, and F. Peri, “Machine Reasoning about Anomalous Sensor Data,” *Ecol. Inform.*, vol. 5, no. 1, pp. 9–18, Jan. 2010.

[16] D. J. Russomanno, C. Kothari, and O. A. Thomas, “Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models,” in *Proceedings of the 2005 International Conference on Artificial Intelligence*, 2005, vol. 2, pp. 1–7.

[17] D. J. Russomanno, C. Kothari, and O. Thomas, “Sensor Ontologies: from Shallow to Deep Models,” in *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory, SSST*, 2005, pp. 107–112.

[18] C. Goodwin and D. J. Russomanno, “An Ontology-Based Sensor Network Prototype Environment,” in *Fifth International Conference on Information Processing in Sensor Networks*, 2006, pp. 1–2.

[19] J.-H. Kim, H. Kwon, D.-H. Kim, H.-Y. Kwak, and S.-J. Lee, “Building a Service-

Oriented Ontology for Wireless Sensor Networks," in *Seventh IEEE/ACIS International Conference on Computer and Information Science (ICIS 2008)*, 2008, pp. 649–654.

[20] R. Jurdak, C. V. Lopes, and P. Bald, "A Framework for Modeling Sensor Networks," in *Proceedings of the Building Software for Pervasive Computing Workshop at OOPSLA*, 2004.

[21] M. Eid, R. Liscano, and A. Saddik, "A Novel Ontology for Sensor Networks Data," in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2006, pp. 75–79.

[22] M. Eid, R. Liscano, and A. el Saddik, "A Universal Ontology for Sensor Networks Data," in *2007 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2007, pp. 59–62.

[23] S. Avancha and C. Patel, "Ontology-driven adaptive sensor networks," in *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS*, 2004, pp. 194–202.

[24] M. Presser, P. Barnaghi, M. Eurich, and C. Villalonga, "The SENSEI project: Integrating the Physical World with the Digital World of the Network of the Future," *IEEE Commun. Mag.*, vol. 47, no. 4, pp. 1–4, Apr. 2009.

[25] S. Kolozali, M. Bermudez-Edo, D. Puschmann, F. Ganz, and P. Barnaghi, "A Knowledge-Based Approach for Real-Time IoT Data Stream Annotation and Processing," in *IEEE International Conference on Internet of Things(iThings), and IEEE Green Computing and Communications (GreenCom), and IEEE Cyber, Physical and Social Computing (CPSCom)*, 2014, pp. 215–222.

[26] H. van der Schaaf and R. Herzog, "Mapping the OGC Sensor Things API onto the OpenIoT Middleware," in *Interoperability and Open-Source Solutions for the Internet of Things*, 2015, pp. 62–70.

[27] M. Leggieri, C. von der Weth, and M. Serrano, "Deliverable 3.1.2: Semantic Representations of Internet-Connected Objects," 2013.

[28] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud, "Standardizing Generic Cross-Domain Applications in Internet of Things," in *IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 589–594.

[29] A. Gyrard and C. Bonnet, "Semantic Web Best Practices: Semantic Web Guidelines for Domain Knowledge Interoperability to Build the Semantic Web of Things," Apr. 2014.

[30] A. Gyrard, C. Bonnet, and K. Boudaoud, "Domain Knowledge Interoperability to Build the Semantic Web of Things," in *W3C Workshop on the Web of Things*, 2014, pp. 1–5.

[31] R. Petrolo, V. Loscri, and N. Mitton, "Towards a SmartCity based on Cloud of Things, a Survey on the Smart City Vision and Paradigms," *Trans. Emerg. Telecommun. Technol.*, pp. 1–11, 2015.

[32] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the Internet of Things," in *Proceedings of the 8th Middleware Doctoral Symposium on - MDS '11*, 2011, pp. 1–6.

[33] N. Seydoux, M. Ben Alaya, N. Hernandez, T. Monteil, and O. Haemmerle, "Semantique et Internet des objets : d'un etat de l'art a une ontologie modulaire," in *26es Journees francophones d'Ingenierie des Connaissances*, 2015.

[34] A. Gyrard, C. Bonnet, and K. Boudaoud, "Helping IoT Application Developers with Sensor-Based Linked Open Rules," in *7th International Workshop on Semantic Sensor Networks in Conjunction with the 13th International Semantic Web Conference (ISWC 2014)*, 2014, pp. 1–4.

[35] oneM2M, “TS-0012 oneM2M Base Ontology,” 2016.

[36] L. Daniele, F. ran. den Hartog, and J. Roes, “Study on Semantic Assets for Smart Appliances Interoperability,” 2015.

[37] UPnP, “UPnP device Architecture 1.1,” 2008.

[38] A. Willner, C. Papagianni, M. Giatili, P. Grosso, M. Morsey, Y. Al-Hazmi, and I. Baldin, “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 {{}{&}{}} Communities (TRIDENTCOM)*, 2015, p. 10.

[39] M. Morsey, A. Willner, R. Loughnane, M. Giatili, C. Papagianni, I. Baldin, P. Grosso, and Y. Al-Hazmi, “DBcloud: Semantic Dataset for the Cloud,” in *Proc. of the 3rd International Workshop on Computer and Networking Experimental Research Using Testbeds (CNERT)*, 2016.

[40] D. Bernstein, V. Deepak, and R. Chang, “Draft Standard for Intercloud Interoperability and Federation (SIIF),” Sep. 2015.

[41] B. Di Martino, G. Cretella, A. Esposito, A. Willner, A. Alloush, D. Bernstein, D. Vij, and J. Weinman, “Towards an Ontology-Based Intercloud Resource Catalogue -- The IEEE P2302 Intercloud Approach for a Semantic Resource Exchange,” in *International Conference on Cloud Engineering*, 2015, pp. 458–464.

[42] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, “Future Internet Research and Experimentation,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, p. 89, Jul. 2007.

[43] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, “GENI: A Federated Testbed for Innovative Network Experiments,” *Comput. Networks*, vol. 61, no. 3, pp. 5–23, Mar. 2014.

[44] J. van der Ham, F. Dijkstra, R. Lapacz, and J. Zurawski, “GFD.206: Network Markup Language Base Schema,” 2013.

[45] M. Ghijsen, J. van der Ham, P. Grosso, and C. de Laat, “Towards an Infrastructure Description Language for Modeling Computing Infrastructures,” in *10th International Symposium on Parallel and Distributed Processing with Applications*, 2012, pp. 207–214.

[46] M. Ghijsen, J. van der Ham, P. Grosso, C. Dumitru, H. Zhu, Z. Zhao, and C. de Laat, “A Semantic-Web Approach for Modeling Computing Infrastructures,” *Comput. Electr. Eng.*, vol. 39, no. 8, pp. 2553–2565, 2013.

[47] J. van der Ham, J. Stéger, S. Laki, Y. Kryftis, V. Maglaris, and C. de Laat, “The NOVI Information Models,” *Futur. Gener. Comput. Syst.*, vol. 42, no. 0, pp. 64–73, 2015.

[48] J. R. Hobbs and F. Pan, “Time Ontology in OWL,” Sep. 2006.

[49] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner, “An Internet of Things Platform for Real-World and Digital Objects,” *Scalable Computing: Practice and Experience*, vol. 13, no. 1, 2012.

[50] S. A. McIlraith, T. C. Son, and Honglei Zeng, “Semantic Web Services,” *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 46–53, Mar. 2001.

[51] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, “Human Mobility Models for Opportunistic Networks,” *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 157–165, Dec. 2011.

[52] FIESTA-IoT, “Deliverable 5.1: Experiments Design and Specification.”

[53] A. Gangemi and V. Presutti, "Ontology Design Patterns," in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 221–243.

[54] E. Simperl, "Reusing ontologies on the Semantic Web: A feasibility study," *Data Knowl. Eng.*, vol. 68, no. 10, pp. 905–925, Oct. 2009.

[55] M. C. Suarez-Figueroa, "NeOn Methodology for Building Networks Ontology: Specification, Scheduling and Reuse," UPM, 2010.

[56] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," 2001.

[57] O. Corcho, M. Fernandez-Lopez, and A. Gómez-Pérez, "Methodologies, Tools and Languages for Building Ontologies. Where is Their Meeting Point?," *Data Knowl. Eng.*, vol. 46, no. 1, pp. 41–64, Jul. 2003.

[58] Y. Sure, S. Staab, and R. Studer, "On-To-Knowledge Methodology (OTKM)," in *Handbook on Ontologies*, S. Staab and R. Stude, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 117–132.

[59] FIESTA-IoT, "Deliverable 2.1: Stakeholders Requirements."

[60] IoT-A, "IoT-A Terminology." [Online]. Available: http://www.iot-a.eu/public/terminology/copy_of_term. [Accessed: 05-Apr-2016].

[61] C. Tejo-Alonso, D. Berrueta, L. Polo, and S. Fernández, "Metadata for Web Ontologies and Rules: Current Practices and Perspectives," in *Metadata and Semantic Research*, Springer Berlin Heidelberg, 2011, pp. 56–67.

[62] S. Peroni, D. Shotton, and F. Vitali, "The Live OWL Documentation Environment: A Tool for the Automatic Generation of Ontology Documentation," in *Knowledge Engineering and Knowledge Management*, 2012, pp. 398–412.

[63] FIESTA-IoT, "Deliverable 2.2: Analysis of IoT Platforms and Testbeds," 2015.

[64] M. Baglioni, F. de Macedo, J. Antonio, C. Renso, R. Trasarti, and M. Wachowicz, "Towards Semantic Interpretation of Movement Behavior," in *Advances in GIScience*, 2009, pp. 271–288.

[65] Y. Hu, K. Janowicz, D. Carral, S. Scheider, W. Kuhn, G. Berg-Cross, P. Hitzler, M. Dean, and D. Kolas, "A Geo-ontology Design Pattern for Semantic Trajectories," in *Spatial Information Theory*, Scarborough, North Yorkshire, UK, 2013, pp. 438–456.

[66] C. Renso, M. Baglioni, J. A. F. de Macedo, R. Trasarti, and M. Wachowicz, "How You Move Reveals Who You Are: Understanding Human Behavior by Analyzing Trajectory Data," *Knowl. Inf. Syst.*, vol. 37, no. 2, pp. 331–362, Nov. 2013.

APPENDIX I – OPENIOT UTILITY METRICS

Following we list the utility metrics in details. The following text is taken from [27]

Utility Metrics for Physical Sensors

- Quality: determines the accuracy and sensitivity of the measurements provided by a sensor and it may also influence energy consumption
  - ssn:Accuracy a owl:Class ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
  - ssn:Sensitivity a owl:Class ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Energy consumption: associated with the lifetime of the sensor network and therefore this metric can be used for functions like accounting, resource optimization and billing
  - spt:Energy a owl:Class ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Bandwidth: refers to a bit-rate measure, representing the available or consumed data communication resources expressed in bits per second or multiples of it (bit/s, kbit/s, Mbit/s, Gbit/s, etc.)
  - openiot:Bandwidth rdfs:subClassOf ssn:MeasurementProperty ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Data volume: volume of data (amount of data) produced by a sensor
  - openiot:DataVolume rdfs:subClassOf ssn:MeasurementProperty ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Trustworthiness: measure specifying that a sensor will deliver true measurements on time within the scope of its technical parameters
  - openiot:Trustworthiness rdfs:subClassOf ssn:MeasurementProperty ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .

Utility Metrics for Virtual Sensors

- Data volume: (as defined above)
- Bandwidth: (as defined above)
- Time of the usage session: time during which a sensor has been used
  - openiot:SessionTime rdfs:subClassOf ssn:observationResultTime ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Virtual sensor location: location of a sensor
  - openiot:VirtualSensorLocation rdfs:subClassOf dul:Location ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Virtual sensor task: part of a business process a sensors serves for
  - openiot:VirtualSensorTask rdfs:subClassOf dul:Process ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Number of physical sensors: number of physical sensor which together represent a virtual sensor
  - openiot:subSensorsCount a owl:DatatypeProperty ;
    rdfs:domain openiot:VirtualSensor ;
    rdfs:range xsd:integer ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Type of physical sensors: Each new type of physical sensor will be defined as a subclass of ssn:Sensor as need be
  - openiot:WindSensor rdfs:subClassOf ssn:Sensor ;
    rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .

- User-defined costs: cost or utility value defined assigned to a sensor assigned by a user
    - openiot:SensorUtility rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .

Utility Metrics for a Sensor Network and Application Service Level

- System lifetime: determines the longevity of sensors
    - ssn:SystemLifetime a owl:Class ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Latency: measures the time delay experienced in a system
    - ssn:Latency a owl:Class ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Quality: quality of a sensor network determined by the quality of the data
    - spt:NetworkQuality a owl:Class ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Delay and delay variation: refer to delay in data collections from sensors
    - openiot:Delay rdfs:subClassOf ssn:MeasurementProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
    - openiot:DelayRange rdfs:subClassOf ssn:MeasurementProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Bandwidth, capacity, throughput: basic parameters determining the performance of sending data over a link within a given time
    - openiot:NetworkBandwidth rdfs:subClassOf spt:NetworkProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
    - openiot:NetworkCapacity rdfs:subClassOf spt:NetworkProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
    - openiot:NetworkThroughput rdfs:subClassOf spt:NetworkProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Hop count: costs of a communication path
    - openiot:LinkCost rdfs:subClassOf spt:LinkProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Ease of deployment: how easy it is to deploy
    - openiot:EaseOfDeployment rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Reliability: ability of the system or components to perform its functions
    - openiot:Reliability rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Survivability: ability of a system or subsystem to continue to function during and after disturbances
    - openiot:Survivability rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Scalability: ability of the system accommodate growth in the number of sensors
    - openiot:Scalability rdfs:subClassOf spt:NetworkProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Resource optimization and cost efficiency: ability of the system to optimize resource utilization.
    - opeiot:ResourceOptimization rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
    - openiot:CostEfficiency rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .
- Relevance: usefulness of an answer to a specific query
    - openiot:Relevance rdfs:subClassOf ssn:OperatingProperty ;
      rdfs:isDefinedBy <https://openiot.eu/ontology/ns> .

- Confidentiality: ability of a system to protect privacy
  - `openiot:Confidentiality rdfs:subClassOf ssn:OperatingProperty ; rdfs:isDefinedBy <http://openiot.eu/ontology/ns> .`

APPENDIX II – "IN-HOUSE" TESTBED QUANTITY KINDS AND UNITS

Below we tabulate different types of QuantityKinds (QK) available with different in-house Testbeds along with the units. The tables also contains M3-lite equivalent QKs and units for the Testbed defined QKs and units.

Table 7: SmartSantander Testbed QuantityKinds and Units

SmartSantander QK	M3-lite QK	Unit	M3-lite Unit
activePower	ActivePower	watt	Watt
atmosphericPressure	AtmosphericPressure	bar	Bar
batteryLevel	BatteryLevel	percent	Percent
chemicalAgentAtmosphericConcentration:03	ChemicalAgentAtmosphericConcentration03	microgramPerCubicMetre	MicrogramPerCubicMetre
chemicalAgentAtmosphericConcentration:airParticles	ChemicalAgentAtmosphericConcentrationAirParticles	milligramPerCubicMetre	MilligramPerCubicMetre
chemicalAgentAtmosphericConcentration:CO	ChemicalAgentAtmosphericConcentrationCO	scale	Scale
chemicalAgentAtmosphericConcentration:NO2	ChemicalAgentAtmosphericConcentrationNO2	microgramPerCubicMetre	MicrogramPerCubicMetre
direction:heading	DirectionHeading	index	Index
electricCurrent	ElectricCurrent	ampere	Ampere
electricField:1800mhz	ElectricField1800mhz	millivoltPerMetre	MillivoltPerMetre
electricField:2100mhz	ElectricField2100mhz	millivoltPerMetre	MillivoltPerMetre
electricField:2400mhz	ElectricField2400mhz	millivoltPerMetre	MillivoltPerMetre
electricField:900mhz	ElectricField900mhz	millivoltPerMetre	MillivoltPerMetre
electricPotential	ElectricPotential	volt	Volt
fillLevel:gasTank:1	FillLevelGasTank:1	percent	Percent
fillLevel:gasTank:2	FillLevelGasTank:2	percent	Percent
fillLevel:wasteContainer	FillLevelWasteContainer	percent	Percent
fuelComsumption:total	FuelComsumptionTotal	litre	Litre
fuelConsumption:instantaneous	FuelConsumptionInstantaneous	litrePer100Kilometres	LitrePer100Kilometres
illuminance	Illuminance	lux	Lux
mass	Mass	kilogram	Kilogram
mileage:distanceToService	MileageDistanceToService	kilometre	Kilometre
mileage:total	MileageTotal	metre	Metre

motionState:vehicle	MotionStateVehicle	index	Index
position:altitude	PositionAltitude	metre	Metre
position:latitude	PositionLatitude	degreeAngle	DegreeAngle
position:longitude	PositionLongitude	degreeAngle	DegreeAngle
presenceState:driverCard:1	PresenceStateDriverCard1	index	Index
presenceState:driverCard:2	PresenceStateDriverCard2	index	Index
presenceState:parking	PresenceStateParking	index	Index
rainfall	Rainfall	millimetrePerHour	MillimetrePerHour
reactivePower	ReactivePower	var	Var
relativeHumidity	RelativeHumidity	percent	Percent
roadOccupancy	RoadOccupancy	percent	Percent
rotationalSpeed:engine	RotationalSpeed:engine	revolutionPerMinute	RevolutionPerMinute
soilMoistureTension	SoilMoistureTension	centibar	Centibar
solarRadiation:par	SolarRadiationPar	wattPerSquareMetre	WattPerSquareMetre
soundPressureLevel:ambient	SoundPressureLevelAmbient	decibelA	DecibelA
speed:average	SpeedAverage	kilometrePerHour	KilometrePerHour
speed:instantaneous	SpeedInstantaneous	metrePerSecond	MetrePerSecond
speed:median	SpeedMedian	kilometrePerHour	KilometrePerHour
temperature:ambient	TemperatureAmbient	degreeCelsius	DegreeCelsius
temperature:engine	TemperatureEngine	degreeCelsius	DegreeCelsius
temperature:soil	TemperatureSoil	degreeCelsius	DegreeCelsius
temperature:wasteContainer	TemperatureWasteContainer	degreeCelsius	DegreeCelsius
timeRelatedState:driver:1	TimeRelatedStateDriver:1	index	Index
timeRelatedState:driver:2	TimeRelatedStateDriver:2	index	Index
timestamp	Timestamp	dimensionless	Dimensionless
trafficCongestion	TrafficCongestion	index	Index
trafficIntensity	TrafficIntensity	vehiclePerMinute	VehiclePerMinute
vehicleOverspeedState	VehicleOverspeedState	index	Index
windDirection	WindDirection	degreeAngle	DegreeAngle
windSpeed	WindSpeed	kilometrePerHour	KilometrePerHour
workingState:driver:1	WorkingStateDriver1	index	Index

workingState:driver:2	WorkingStateDriver2	index	Index

Table 8: Com4Innov's Testbed QuantityKinds and Units

Com4Innov Testbed QK	M3-lite QK	Unit	M3-lite Unit
AirTemperature	TemperatureAmbient	Celsius	DegreeCelsius
BoardTemperature	TemperatureBoard	Celsius	DegreeCelsius
Delta Dewpoint	DeltaDewPoint	Natural Number	Index
Dewpoint	DewPoint	Celsius	DegreeKelvin
OutputVoltage	ElectricPotential	Volt	Volt
Radiation	SolarRadiation	W/m^2	WattPerSquareMetre
RelativeHumidity	RelativeHumidity	Percentage	Percent
SoundLevel	SoundPressureLevelAmbient	dB	Decibel

Table 9: KETI's Testbed QuantityKinds and Units

KETI's Testbed QK	M3-lite QK	Unit	M3-lite Unit
Office Temperature	BuildingTemperature	Celsius	DegreeCelsius
Office Humidity	RelativeHumidity	Percentage	Percent
Office Illumination	Illuminance	Lux	Lux
Office CO2	ChemicalAgentAtmosphericConcentrationCO2	ppm	PPM
Electricity	ActivePower	Watt	Watt
Parking	PresenceStateParking	Boolean	Index
User Occupancy	PresenceStatePeople	Boolean	Index

Table 10: UNIS's Testbed QuantityKind and Units

UNIS's Testbed QK	M3-lite QK	Unit	M3-lite Unit
Temperature	BuildingTemperature	Celsius	DegreeCelius
Office Humidity	RelativeHumidity	Percentage	Percent
Office Illumination	Illuminance	Lux	Lux
Office CO2	ChemicalAgentAtmosphericConcentrationCO2	ppm	PPM
Electricity	ActivePower	Watt	Watt
User presence	PresenceStatePeople	Boolean	Index

APPENDIX III – SAMPLE USAGE OF ONTOLOGY

In this appendix we provide sample annotations generated by SmartSantander Testbed and UNIS's Testbed for resources as well as observations.

Sample annotation of SmartSantander Testbed Resources

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
@prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix sms-srd: <http://api.smartsantander.eu#> .
@prefix fiesta-iot-srd-href: <http://platform.fiesta-iot.eu/srd/registry/poc/> .

sms-srd:SmartSantanderTestbed a ssn:Deployment .

sms-srd:platform.urn:x-iot:smartsantander:u7jcfa:t10000 a ssn:Platform ;
       geo:location sms-srd:location.urn:x-iot:smartsantander:u7jcfa:t10000 .

sms-srd:location.urn:x-iot:smartsantander:u7jcfa:t10000 a geo:Point ;
       geo:lat "43.47171"^^xsd:float ;
       geo:long "-3.80014"^^xsd:float .

fiesta-iot-srd-href:urn:x-iot:smartsantander:u7jcfa:t10000 a ssn:Device ;
       ssn:hasDeployment sms-srd:SmartSantanderTestbed ;
       ssn:hasSubSystem fiesta-iot-srd-href:urn:x-
iot:smartsantander:u7jcfa:t10000.BatteryLevel.Sensor , fiesta-iot-srd-href:urn:x-
iot:smartsantander:u7jcfa:t10000.AirTemperature.Sensor , fiesta-iot-srd-href:urn:x-
iot:smartsantander:u7jcfa:t10000.Illuminance.Sensor ;
       ssn:onPlatform sms-srd:platform.urn:x-iot:smartsantander:u7jcfa:t10000 .

fiesta-iot-srd-href:urn:x-iot:smartsantander:u7jcfa:t10000.BatteryLevel.Sensor a
ssn:SensingDevice ;
       iot-lite:exposedBy sms-srd:service.urn:x-
iot:smartsantander:u7jcfa:t10000.BatteryLevel.Sensor ;
       iot-lite:hasQuantityKind m3-lite:BatteryLevel ;
       iot-lite:hasUnit m3-lite:Percent .

fiesta-iot-srd-href:urn:x-iot:smartsantander:u7jcfa:t10000.AirTemperature.Sensor a
ssn:SensingDevice ;
       iot-lite:exposedBy sms-srd:service.urn:x-
iot:smartsantander:u7jcfa:t10000.AirTemperature.Sensor ;
       iot-lite:hasQuantityKind m3-lite:AirTemperature ;
       iot-lite:hasUnit m3-lite:DegreeCelsius .

fiesta-iot-srd-href:urn:x-iot:smartsantander:u7jcfa:t10000.Illuminance.Sensor a
ssn:SensingDevice ;
       iot-lite:exposedBy sms-srd:service.urn:x-
iot:smartsantander:u7jcfa:t10000.Illuminance.Sensor ;
       iot-lite:hasQuantityKind m3-lite:Illuminance ;
       iot-lite:hasUnit m3-lite:Lux .

sms-srd:service.urn:x-iot:smartsantander:u7jcfa:t10000.BatteryLevel.Sensor a iot-
lite:Service ;
       iot-lite:endpoint "https://api-
dev.smartsantander.eu:10443/v2/measurements/batteryLevel/urn/urn:x-
iot:smartsantander:u7jcfa:t10000/last?format=jsonld"^^xsd:string ;
       iot-lite:type "REST"^^xsd:string .
```



```
sms-srd:service.urn:x-iot:smartsantander:u7jcfa:t10000.AirTemperature.Sensor a iot-
lite:Service ;
       iot-lite:endpoint "https://api-
dev.smartsantander.eu:10443/v2/measurements/temperature:ambient/urn/urn:x-
iot:smartsantander:u7jcfa:t10000/last?format=jsonld"^^xsd:string ;
       iot-lite:type "REST"^^xsd:string .

sms-srd:service.urn:x-iot:smartsantander:u7jcfa:t10000.Illuminance.Sensor a iot-
lite:Service ;
       iot-lite:endpoint "https://api-
dev.smartsantander.eu:10443/v2/measurements/illuminance/urn/urn:x-
iot:smartsantander:u7jcfa:t10000/last?format=jsonld"^^xsd:string ;
       iot-lite:type "REST"^^xsd:string .
```

Annotations of Sample Data from the SmartSantander Testbed

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
@prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix fiesta-iot-srd-href: <http://platform.fiesta-iot.eu/srd/registry/poc/> .

_:b0 a ssn:Observation ;
       ssn:observationResult _:b1 ;
       ssn:observationSamplingTime _:b2 ;
       ssn:observedBy fiesta-iot-srd-href:urn:x-
iot:smartsantander:u7jcfa:t10000.AirTemperature.Sensor ;
       ssn:observedProperty m3-lite:AirTemperature ;
       geo:location _:b3 .

_:b1 a ssn:SensorOutput ;
       ssn:hasValue _:b4 .

_:b2 a dul:TimeInterval ;
       dul:hasIntervalDate "Fri Apr 01 2016 08:32:34 GMT+0200 (CEST)"^^xsd:dateTime .

_:b3 a geo:Point ;
       geo:lat "-3.80014E0"^^xsd:float ;
       geo:long "4.347171E1"^^xsd:float .

_:b4 a ssn:ObservationValue ;
       iot-lite:hasUnit m3-lite:DegreeCelsius ;
       dul:hasDataValue "2.283E1"^^xsd:float .
```

Annotation of Sample Data from the UNIS's Testbed

```
@prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#> .
@prefix qu: <http://purl.org/NET/ssnx/qu/qu#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix fiesta-res: <http://platform.fiesta-iot.eu/srd/registry/poc/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix sc: <http://iot.ee.surrey.ac.uk/smartcampus#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix qu-rec20: <http://purl.org/NET/ssnx/qu/qu-rec20#> .
@prefix dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#> .
```



```
@prefix time: <http://www.w3.org/2006/time#> .

sc:UK-GU-UNIS-ICS-Desk1 a geo:Point ;
       geo:lat "51.24335593858872"^^xsd:string ;
       geo:long "-0.5931907234437758"^^xsd:string .

sc:urn:x-iot:smartcampus:smart-ics:obs001.presence ssn:obserationSamplingTime sc:urn:x-
iot:smartcampus:smart-ics:ti001.timestamp ;
       ssn:observationResult sc:urn:x-iot:smartcampus:smart-ics:so001.presence ;
       ssn:observedBy fiesta-res:urn:x-iot:smartcampus:smart-ics:n001.presence ;
       ssn:observedProperty m3-lite:Presence ;
       geo:location sc:UK-GU-UNIS-ICS-Desk1 .

sc:urn:x-iot:smartcampus:smart-ics:ti001.timestamp dul:hasIntervalDate "2016-03-
18T08:29:47.227Z"^^xsd:string .

sc:urn:x-iot:smartcampus:smart-ics:so001.presence ssn:hasValue sc:urn:x-
iot:smartcampus:smart-ics:obsval001.presence .

fiesta-res:urn:x-iot:smartcampus:smart-ics:n001.presence a ssn:SensingDevice .

sc:urn:x-iot:smartcampus:smart-ics:obsval001.presence iot-lite:hasUnit m3-lite:Boolean ;
       dul:hasDataValue "0"^^xsd:string .

m3-lite:Boolean a qu:Unit .
```

FIESTA 2016