RESPOND

DEMAND RESPONSE FOR ALL

Integrated Demand REsponse SOlution Towards Energy POsitive NeighbourhooDs

WP 4 – ICT enabled cooperative Demand Response model

T4.1: Semantic Information Model

D4.1 Semantic Information Model

The RESPOND Consortium 2019

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 768619

PROJECT ACRONYM	RESPOND
DOCUMENT	**D4.1 Semantic Information Model**
TYPE (DISTRIBUTION LEVEL)	☑ Public ☐ Confidential ☐ Restricted
DELIVERY DUE DATE	31/03/2019
DATE OF DELIVERY	31/03/2019
STATUS AND VERSION	1.2
DELIVERABLE RESPONSIBLE	TEK
AUTHOR (S)	Iker Esnaola (TEK) Francisco Javier Diez (TEK) Nikola Tomasevic (IMP) Marko Batic (IMP)
OFFICIAL REVIEWER(S)	Nikola Tomasevic (IMP)

DOCUMENT HISTORY

	ISSUE DATE	CONTENT AND CHANGES
V0.1	10/01/2019	Table of content
V1.0	06/03/2019	First version by TEK
V1.1	22/03/2019	Contributions from IMP
V1.2	27/03/2019	Review by IMP

EXECUTIVE SUMMARY

This deliverable presents the semantic information model developed for the RESPOND project. The aim of this model is to provide resources with an explicit and unambiguous semantics. Without this explicit semantic assignment, different users could refer to the same resource with different meanings. Furthermore, a semantic information model improves semantic interoperability, providing both humans and machines with a shared meaning of terms.

The proposed semantic model is based on the RESPOND ontology, which has been developed based on the well-known NeOn Methodology to ensure its quality. Moreover, a set of related ontologies have been reviewed and the most adequate have been reused. On top of that, the ontology has been made online and it is well-documented, which is expected to further foster its understandability and reusability.

The ontology has been then instantiated with information of buildings and installed devices. This information is extracted from a set of Excel sheets filled by people in charge of the RESPOND pilot sites. However, the measurements and actuations made by IoT systems is not used to instantiate the ontology. Instead, this information is stores in InfluxDB and the ontology instantiation contains the Data Point ID.

Although an RDF Store with sufficient inferencing capabilities was not found, RESPOND proposes an approach that provides reasoning in real-time, avoiding the storage of undesirable outdated inferences. A Virtuoso Open Source RDF Store is used to host the semantic information, but the proposed approach could be easily extended to work with other RDF Stores.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

AEC	Architecture Engineering & Construction
BMS	Building Management System
BOT	Building Topology Ontology
CDM	Canonical Data Model
DUL	DOLCE ultra-lite
DR	Demand Response
EMS	Energy Management System
ETSI	European Telecommunications Standards Institute
GIS	Geographic Information Systems
HVAC	Heating, Ventilation and Air Conditioning
IDE	Intelligent Domotic Environment
IoT	Internet of Things
ODP	Ontology Design Pattern
OEG	Ontology Engineering Group
ORSD	Ontology Requirements Specification Document
SAREF	Smart Appliances REFerence
SEAS	Smart Energy Aware Systems
SOSA	Sensor, Observation, Sample, and Actuator
SSN	Semantic Sensor Network
SSO	Stimulus-Sensor-Observation
URI	Uniform Resource Identifier
WIDOCO	Wizard for DOCumenting Ontologies
W3C	World Wide Web Consortium

1. INTRODUCTION

1.1 AIMS AND OBJECTIVES

This task deals with the **integration of all the data** managed by heterogeneous systems and tools used to manage the energy dispatching and environmental conditions. More specifically, the development of a semantic model is aimed at providing a shared understanding of data, thus avoiding semantic interoperability issues.

Furthermore, the exploitation of Semantic Technologies is expected to pave the way for other services that could exploit data's underlying meaning for enhanced results. This includes analytic services, which many times fail to get optimal outputs due to their inability to take leverage of the inherent semantics of the data being used to develop predictive models.

As for the development of the semantic model, one of its aims is to follow good practices of reuse of existing ontologies as much as possible. Furthermore, the instantiation of this semantic model and the data contained in the semantic data repository is not aimed to be static. Instead, the same services that exploit these data are expected also to contribute in generating new data and enrich this repository.

1.2 RELATION TO OTHER PROJECT ACTIVITIES

With regards to the interaction between Task 4.1 and the rest of RESPOND project activities, the main interactions are listed below:

- As for the WP4, the T4.1 supports T4.2 (Integration of demand response with supply/demand side management), T4.3 (Optimal energy dispatching at household and neighbourhood level), T4.4 (Energy production and demand forecasting) and T4.5 (Data analytics and optimized control) in order to provide semantic enrichment of data and relationships between physical and statistical concepts thus supporting their internal processes. These tasks will leverage the semantic repository developed in T4.1 to exploit data more efficiently and, likewise, the rest of the tasks are expected to further extend the semantic repository.
- As for the WP5, T4.1 and the common vocabulary it defines will contribute in the communication and integration among different key technology tiers of RESPOND solution. Furthermore, T5.4 (Integration with desktop dashboard and smart mobile client) will target the semantic repository to query necessary data and visualize it adequately.

1.3 DELIVERABLE STRUCTURE

The rest of the deliverable is structured as follows. Section 2 introduces Semantic Technologies. This section is not aimed at giving an exhaustive insight of these technologies, but instead a brief introduction to them. Section 3 reviews existing ontologies which are related to the problem at hand. Focus is placed on ontologies that cover the building domain, as well as on ontologies representing observations, actuations, and similar terminology. Section 4 presents the RESPOND ontology. The ontology's development, documentation and instantiation processes are described. Furthermore, the ontology is evaluated from a design correctness point of view and its usage is shown. In section 5 the Semantic Information Repository is shown. Finally, conclusions of this deliverable are presented in section 6.

2. SEMANTIC TECHNOLOGIES

Semantic Technologies provide a single data model for representing any kind of information. They allow adding a logical basis to this representation using different languages. Moreover, they intensively focus on linking diverse graphs of information together in a web-like fashion and allow to combine different representations of information. Therefore, the adoption of Semantic Technologies is aimed to bring advantages such as data integration and interoperability.

The Resource Description Framework (RDF) is a W3C recommendation for representing information on the Web. The basic structure are triples, which consist of a subject, a predicate and an object, as shown in Figure 1. A set of RDF triples constitutes an RDF graph, which can be viewed as node and directed-arc diagrams. Moreover, OWL is a W3C language designed to represent richer and more complex knowledge about things, groups of things, and relations between things.

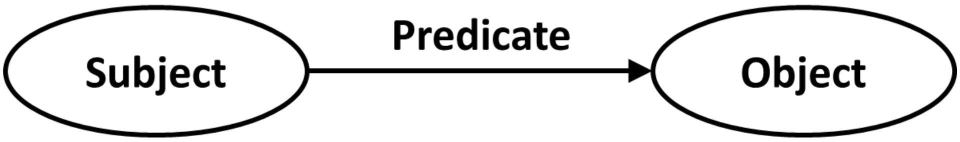


Figure 1: An RDF graph with two nodes (Subject and Object) and a Predicate connecting them.

These resources are described using URIs (Uniform Resource Identifier). It is important to note that RDF is not a data format, but a data model for describing resources. Therefore, although expressing RDF triples as a graph may be convenient to display data, this may not be the most compact or human-friendly way to see the relation between entities. These needs derived in different RDF serialization formats: RDF/XML, RDFa, N-triples and Turtle, among others.

Semantic Technologies also enable inferencing. “Inference” means that automatic procedures can generate new relationships based on the data and based on some additional information in the form of a vocabulary, e.g., a set of rules. Whether the new relationships are explicitly added to the set of data, or are returned at query time, is an implementation issue.

On top of these languages and technologies, the term Linked Data refers to a set of best practices for publishing and interlinking structured data on the Web [1]. These best practices are also known as Linked Data principles, and they can be summarized as follows:

- Use URIs as names for things.
- Use HTTP URIs, so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards.
- Include links to other URIs so that they can discover more things.

Tim Berners-Lee, the inventor of the Web and Linked Data initiator, suggested the 5-star deployment scheme[1] for Linked Open Data shown in Figure 2. The higher the number of stars, the bigger the benefits both for data publishers and users.

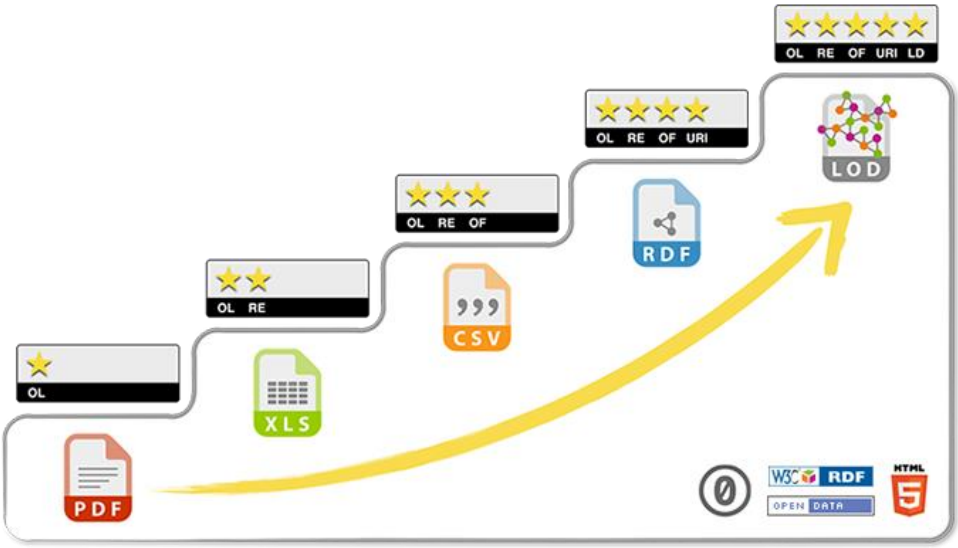


Figure 2: Linked Open Data's 5-star rating system.

Summarizing, the global semantics of RDF allow retrieving the relevant information transparently. URI dereferencing and federated query capabilities allow retrieving information regardless of physical structure and location. RDF provides a natural way to implement a consolidated information space within and across organizations, while allowing re-use of assets such as reference ontologies and vocabularies to standardize datatypes, define types, ranges, etc. In some cases, these ontologies and vocabularies facilitate querying, even when lacking pre-defined and common structures, through common or related (linked) values.

Semantic Technologies can play a critical role in information systems by providing global identifiers, self-describing data, reasoning capabilities and interoperable, machine-understandable format. Therefore, Semantic Technologies are also present in the RESPOND project.

[1] https://5stardata.info

3. Existing ontology review

Nowadays, there is a plethora of ontologies related with the domain of discourse of the RESPOND project. Prior to developing a new ontology or choosing an ontology (or a set of them), it was found necessary to make a review of the most relevant ones. More specifically, the reviewed ontologies are focused on the representation of buildings, the qualities observed, and the devices used to observe them. This review is based on the ontology review presented in [2].

3.1 Building domain ontologies

The representation of buildings constitutes one of the main areas of discourse in RESPOND. Furthermore, recent research projects suggest that the usage of semantic web technologies for managing building data enables the interconnection of information between disciplines in the AEC (Architecture Engineering) domain, which is found to be necessary. Therefore, the building domain has been very active in the last years, with the proposal of different ontologies to satisfy the aforementioned necessity.

ifcOWL

The ifcOWL ontology[2] [3] provides an OWL representation of the EXPRESS schemas of IFC (ISO 16739:2013[3]) for representing building and construction data. Using the ifcOWL ontology, IFC-based building models can be represented as directed labelled graphs. Furthermore, resulting RDF graphs can be linked to related data including material data, GIS (Geographic Information Systems) data or product manufacturer data.

The ifcOWL ontology is a necessary tool to incorporate IFC models to the Semantic Web infrastructure but resulting graphs will be at least as large and complex as the original IFC models. This derives in models that may be too complicated and even inconvenient for some scenarios. Therefore, more light-weighted models are needed.

DogOnt

The DogOnt ontology[4] [4] formalizes IDE (Intelligent Domotic Environment) aspects and it is designed with a particular focus on interoperation between domotic systems. Although it primarily models devices, states and functionalities, DogOnt also supports the description of residential environments where devices are located.

[2] http://ifcowl.openbimstandards.org/IFC4/ADD2.owl
[3] https://www.iso.org/standard/51622.html
[4] http://elite.polito.it/ontologies/dogont.owl

The latest DogOnt version available at the moment of writing this deliverable (version 4.0.1), counts with over 1,000 classes and over 70 properties, which may be rather large in some cases. Furthermore, its scarce documentation may further hinder its usage.

BOT

The Building Topology Ontology[5] (BOT) [5] is a minimal OWL DL ontology for covering core concepts of a building and for defining relationships between their subcomponents. A first design principle for the design of BOT has been to keep a light schema that could promote its reuse as a central ontology in the AEC domain. BOT was developed by W3C's Linked Building Data Community Group.

BOT describes sites comprising buildings, composed of storeys which have spaces that can contain and be bounded by building elements. Sites, buildings, storeys and spaces are all non-physical objects defining a spatial zone. These basic concepts and properties make the schema no more complex than necessary and this design makes the ontology a baseline extensible with concepts and properties from more domain specific ontologies. Therefore, BOT serves as an ontology to be shared. Figure 3 shows the main classes and properties of BOT.

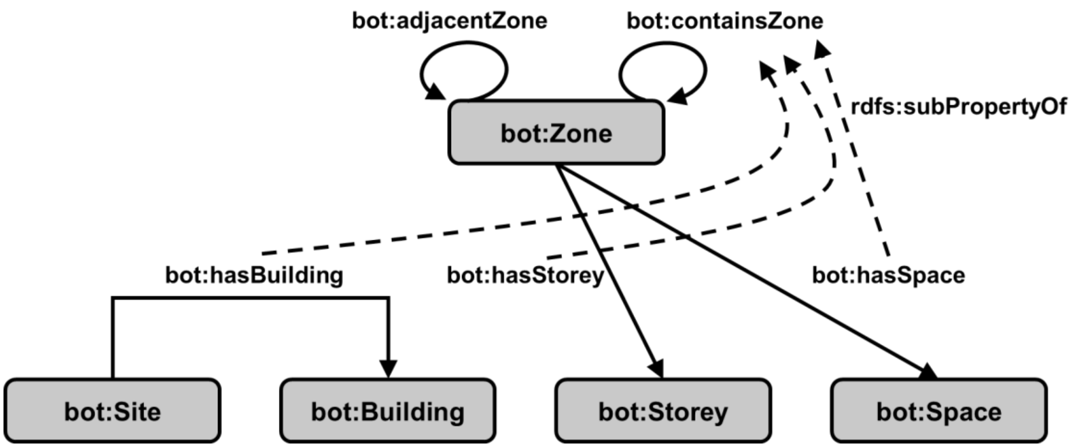


Figure 3: BOT's main classes and properties.

3.2 OBSERVATIONS, ACTUATIONS AND RELATED DOMAIN ONTOLOGIES

The rapid adoption of the IoT leads to an exponential growth of the number of existing devices worldwide. However, the most highlighted drawbacks of the IoT lies in the data level heterogeneity originated from different data models and formats supported by various device manufacturers. Such a diversity derives in semantic interoperability problems, where each system can represent the same thing in different ways,

[5] https://w3id.org/bot

hindering the integration and understanding between these systems. In this regard, ontologies can help alleviating these situations.

SSN

The initial Semantic Sensor Network (SSN) ontology[6] [6] was developed by the W3C Semantic Sensor Networks Incubator Group (SSN-XG) and it proposed a conceptual schema for describing sensors, accuracy and capabilities of such sensors, their observations and methods used for sensing them. Concepts for operating and survival ranges were also included, as well as sensors' performance within those ranges. Finally, a structure for field deployment was defined to describe deployment lifetime and sensing purposes. The initial SSN ontology was aligned with DOLCE ultra-lite (DUL) ontology and built on top of the Stimulus-Sensor-Observation (SSO) Ontology Design Pattern (ODP) describing the relationships between sensors, stimulus, and observations.

Recently, an update of the SSN ontology has been proposed [7] (from now on referred to as SOSA/SSN ontology[7]) that became a W3C recommendation. This new ontology follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called SOSA[8] (Sensor, Observation, Sample, and Actuator) for its elementary classes and properties. Furthermore, the SOSA/SSN ontology's scope is not limited to observations, but it is extended to cover actuations and samplings. In line with the changes implemented in the SOSA/SSN ontology, SOSA drops the direct DUL alignment although it can still be optionally achieved via the SSN-DUL alignment module. Moreover, similar to the original SSO pattern, SOSA acts as a central building block for the new SOSA/SSN ontology but puts more emphasis on its lightweight expressivity and the ability to be used standalone. Then, constraint axioms are added to the vertical module extension named SSN. Figure 4 shows the main SOSA/SSN classes and relationships involved in an Observation.

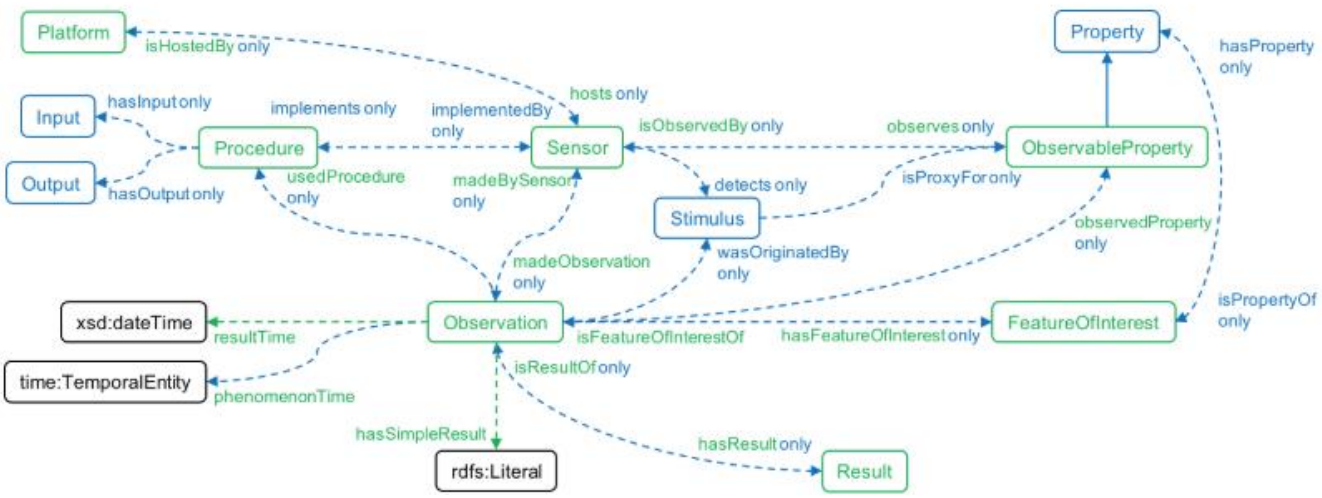


Figure 4: SOSA/SSN's main classes and properties involved in an Observation.

---

[6] http://purl.oclc.org/NET/ssnx/ssn
[7] http://www.w3.org/ns/ssn/
[8] http://www.w3.org/ns/sosa/

Neither the previous SSN ontology nor the new SOSA/SSN ontology describe the different qualities which can be measured by sensors or acted on by actuators. Neither are covered related concepts such as units of measurements of these qualities, hierarchies of sensor/actuator/sampler types, or spatiotemporal terms. All this knowledge has to be modelled by the user, or preferably imported from other existing vocabularies.

SAREF

The Smart Appliances REFerence (SAREF) ontology[9] [8] is a shared model of consensus that facilitates the matching of existing assets in the smart appliances domain. The ontology is standardized by ETSI[10], the European Telecommunications Standards Institute, and provides building blocks that allow the separation and recombination of different parts of the ontology depending on specific needs. The central concept of the ontology is the *saref:Device* class, which is modelled in terms of functions, associated commands, states and provided services. The ontology describes types of devices such as sensors and actuators, white goods, HVAC (Heating, Ventilation and Air Conditioning) systems, lighting and micro renewable home solutions. A device makes an observation (which in SAREF is represented as *saref:Measurement*) which represents the value and timestamp and it is associated with a property (*saref:Property*) and a unit of measurement (*saref:UnitOfMeasure*). The description of these concepts is focused on the residential sector.

The modular conception of the ontology allows the definition of any new device based on building blocks describing functions that devices perform. As previously stated, for the building-related concepts SAREF provides the link to the FIEMSER data model. Furthermore, SAREF can be specialized to refine the general semantics captured in the ontology and create new concepts. The only requirement is that any extension/specialization may comply with SAREF.

One of these specializations is the SAREF4ENER ontology, which focuses on the energy domain. However, at the moment of writing this deliverable, this ontology was inconsistent.

SEAS

The SEAS Ontology[11] [9] is an ontology designed as a set of simple core ODPs that can be instantiated for multiple engineering related verticals. It is an output of the SEAS (Smart Energy Aware Systems) funded by ITEA3, and it is planned to be consolidated with the SAREF ontology as part of ETSI's Special Task Force 556. The SEAS ontology modules are developed based on the following three core modules: the SEAS Feature of Interest ontology[12] which defines features of interest (*seas:FeatureOfInterest*) and their qualities (*seas:Property*), the SEAS Evaluation ontology[13] describing evaluation of these qualities, and the

[9] http://ontology.tno.nl/saref
[10] https://www.etsi.org/
[11] https://w3id.org/seas/
[12] https://w3id.org/seas/FeatureOfInterestOntology
[13] https://w3id.org/seas/EvaluationOntology

SEAS System ontology[14] representing virtually isolated systems connected with other systems. The Procedure Execution (PEP) ontology[15], which is not strictly a SEAS ontology module, but it is contained under the same SEAS project, defines procedure executors that implement procedure methods, and generate procedure execution activities. Furthermore, PEP defines an ODP as a generalization of SOSA's sensor-procedure-observation and actuator-procedure-actuation models.

On top of these core modules, several vertical SEAS ontology modules are defined, which are dependent of a specific domain. Moreover, the SEAS ontology offers a set of alignments to ontologies like SOSA/SSN and QUDT.

[14] https://w3id.org/seas/SystemOntology
[15] https://w3id.org/pep/

4. THE RESPOND ONTOLOGY

This section describes the RESPOND ontology, which is focused on the representation of buildings, their topology and the appliances, devices and other systems installed within them. The RESPOND ontology is available in https://w3id.org/def/respond.

4.1 ONTOLOGY DEVELOPMENT

Ontologies must be carefully designed and implemented, as these tasks have a direct impact on their final quality. Therefore, the use of well-founded ontology development methodologies is advised. For the development of the RESPOND ontology, the NeOn Methodology (shown in Figure 5) was followed mainly because unlike other methodologies it does not prescribe a rigid workflow, but instead it suggests a variety of paths.

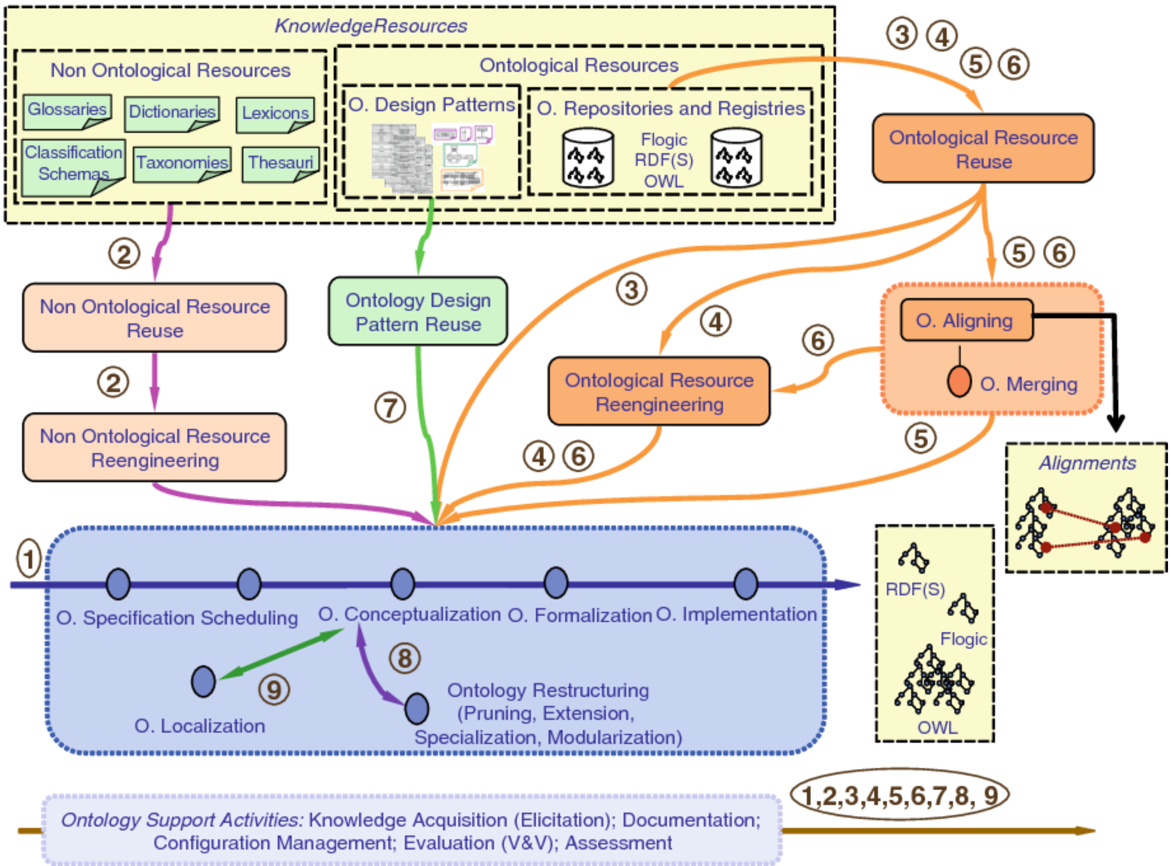


Figure 5: The NeOn Methodology.

These paths are classified as scenarios which consist of different tasks that ontology engineers must follow towards the development of a final ontology that satisfies the tackled problem. Some of the performed tasks during the development of the RESPOND ontology are the ontology reuse, ontology alignment and the modelling of the ontology itself.

4.1.1 Ontology Reuse

The reuse of ontological resources built by others that have already reached some degree of consensus is a good practice in ontology development processes. According to W3C's Data on the Web Best practices[16], the reuse of an existing vocabulary not only captures and facilitates consensus in communities, but also increases interoperability and reduces redundancies. Furthermore, this practice brings other important benefits:

- It increases the quality of the applications reusing ontologies, as these applications become interoperable and they are provided with a deeper, machine processable and commonly agreed-upon understanding of the underlying domain of interest.
- It reduces the costs related to ontology development because it avoids the reimplementation of ontological components, which are already available on the Web and can be directly (or after some additional customization tasks) integrated into a target ontology.
- It potentially improves the quality of the reused ontologies, as these are continuously revised and evaluated by various parties through reuse.

In order to reuse existing ontological resources, the following activities are advised:

1. Ontology Search. This activity consists in finding appropriate ontological resources that meet the requirements described in the ORSD. The existing ontology catalogues such as LOV[17] can ease this task [10].
2. Ontology Assessment. This activity deals with assessing the usability of an ontology with respect to the ontology requirements. This may end up being a laborious task due to the different criteria that may make ontologies suitable for a certain use case. Furthermore, the frequent scarce documentation of ontologies may hinder this activity.
3. Ontology Comparison. In this activity, assessed ontologies should be compared according to criteria that encompass the content of the ontology, the organization of these contents, the language in which it is implemented, the methodology that has been followed to develop it, the software tools used to build and edit the ontology, and the costs of the ontology [11].
4. Ontology Selection. After assessing and comparing ontologies, the most appropriate one or ones (preferably standardized ones) have to be selected in order to reuse them by integrating them in the new ontology being developed.

The RESPOND ontology is developed following the best practices, and it follows the ontology reuse practice. The aforementioned four steps for ontology reuse (ontology search, assessment, comparison and selection) are performed based on the extensive review of existing ontologies. The performance of

[16] https://www.w3.org/TR/dwbp/
[17] http://lov.linkeddata.es

these tasks is expected to increase both the quality of the RESPOND ontology, as well as to ease its understanding and improve its usefulness.

4.1.2 Ontology Alignment

Ontology alignment can be understood as the process of determining correspondences between concepts in ontologies. The explicit alignment of terms from different domain ontologies promotes interoperability, as they facilitate the integration of data represented using different ontologies. Furthermore, the setting mappings to a common upper ontology alleviates integration problems, helps to ensure clarity in modelling and avoids errors that have unintended reasoning implications.

This is why, in order to foster its reusability and interoperability, the RESPOND ontology is aligned to other related domain and upper-level ontologies.

4.1.3 Tools for ontology modelling

Among the available software for building and maintaining ontologies (e.g. PoolParty[18] or TopBraid Composer[19]), Protégé[20] was chosen. Protégé exists in a variety of frameworks (e.g. desktop system or web-based), and in the development of the RESPOND ontology, the Protégé desktop version 5.1.04 was used. Figure 6 shows the visualization of the RESPOND ontology in Protégé.

[18] https://www.poolparty.biz/

[19] https://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/

[20] https://protege.stanford.edu/

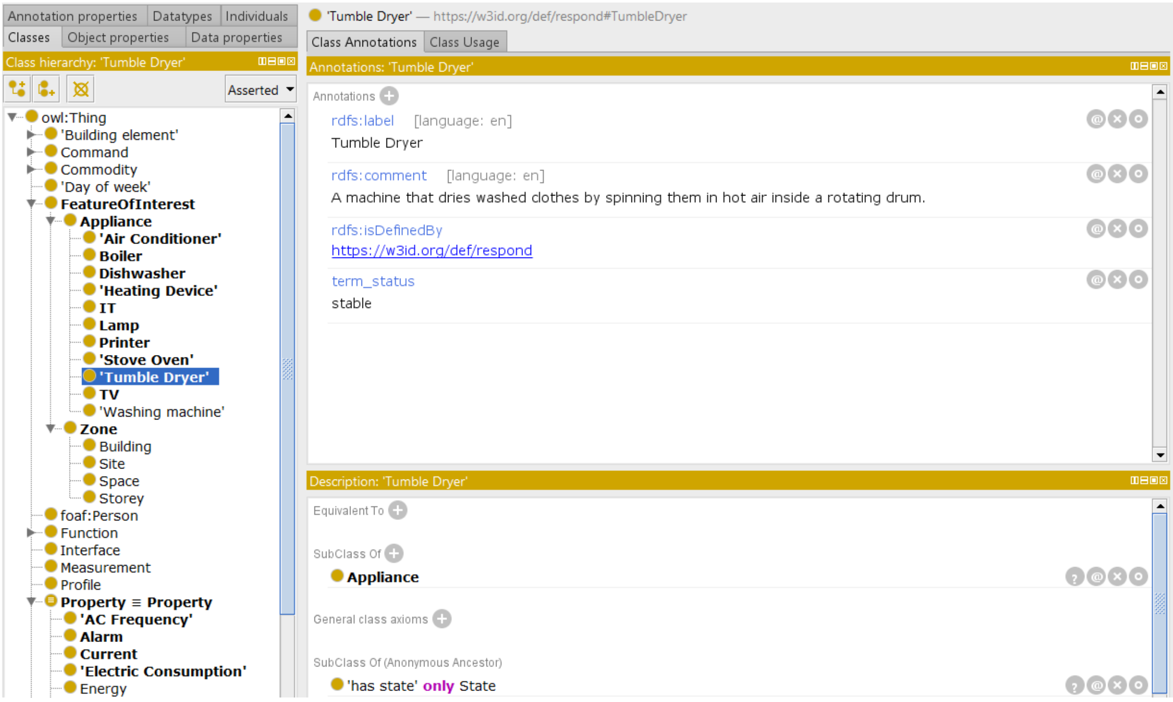


Figure 6: RESPOND ontology's visualization in Protégé.

As for managing the different versions of the ontology, a version control system was necessary. A version control system records changes to a file or set of files over time, so that specific versions can be retrieved later on. The development of the RESPOND ontology was managed with a Git repository, which for transparency purposes, it is available online in https://github.com/RESPOND-PROJECT/RESPOND-Ontology.

4.2 ONTOLOGY REPRESENTATION

The RESPOND ontology reuses BOT, SAREF and SEAS Feature of Interest ontologies, and defines new axioms to address a set of requirements that remain unsatisfied. Although the RESPOND ontology is not aimed at providing semantic annotations of the data gathered by meters, sensors, actuators or other IoT systems, it is necessary to define a property to describe the Data Point ID. This Data Point ID is the identifier that serves as a bridge between an IoT device (represented in the RDF graph) and its measurements (stored in an InfluxDB Time Series Database). This link is described with the *respond:hasDataPointID* data property. Figure 7 shows the RESPOND ontology's main classes and properties.

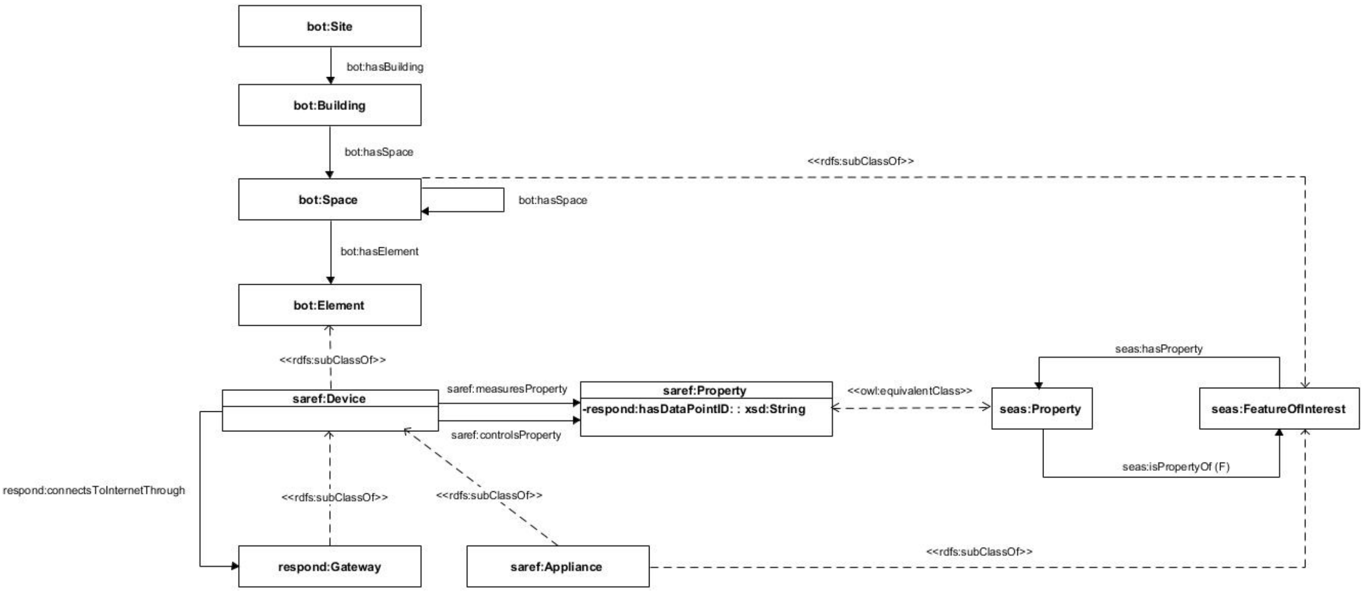
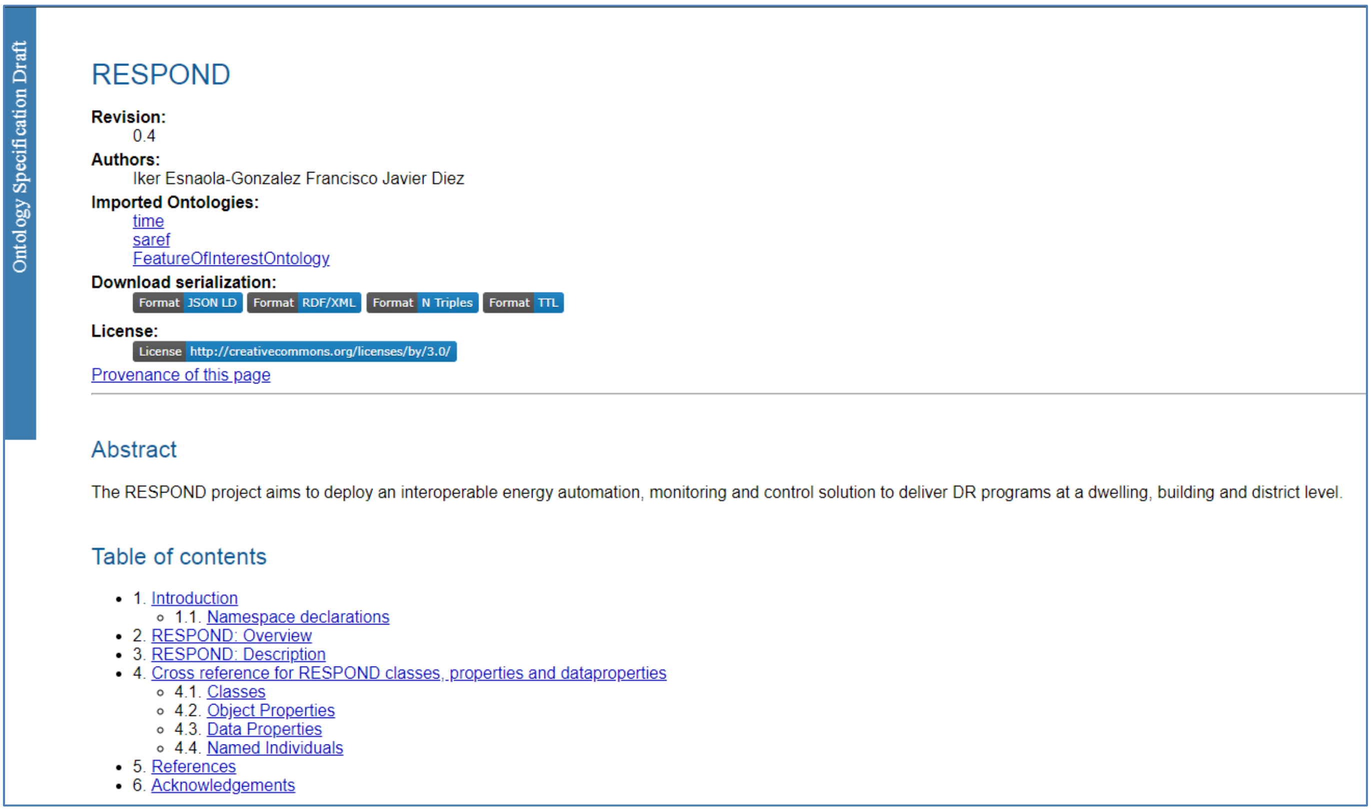


Figure 7: RESPOND Ontology's main classes and properties.

The RESPOND ontology is expected to be extended in further iterations of the project, in order to cover the identified requirements that may arise. For instance, the modelling of Demand Response context such as the interaction between the DR aggregator and the end-user or the pricing of electricity, is foreseen to be essential in the future. However, at the moment of writing this deliverable, details related to DR are yet to be fixed, thus trying to model this knowledge is not worth the effort. Due to this reason, the RESPOND ontology is designed with views to ease its management and update, in order to include more axioms that satisfy future requirements.

4.3 Ontology Documentation

When discovering an ontology, one of the first activities consists in reading its documentation to understand the ontology domain and determine whether it describes this domain appropriately or not. This is why nowadays most ontologies have comprehensive web pages describing their theoretical backgrounds and features. And so it does the RESPOND ontology. The tool used for this aim is WIDOCO (a Wizard for DOCumenting Ontologies) [12], which creates a documentation with diagrams, human readable descriptions of the ontology terms and a summary of changes with respect to previous versions of the ontology. More specifically, the documentation consists of a set of linked enriched HTML pages that can be further extended by users. Figure 8 shows an excerpt of the RESPOND ontology documentation available in https://w3id.org/def/repond .


Ontology Specification Draft

RESPOND

Revision:
0.4
Authors:
Iker Esnaola-Gonzalez Francisco Javier Diez
Imported Ontologies:
time
saref
FeatureOfInterestOntology
Download serialization:
Format JSON LD | Format RDF/XML | Format N Triples | Format TTL
License:
License http://creativecommons.org/licenses/by/3.0/
Provenance of this page

Abstract

The RESPOND project aims to deploy an interoperable energy automation, monitoring and control solution to deliver DR programs at a dwelling, building and district level.

Table of contents

- 1. Introduction
  - 1.1. Namespace declarations
- 2. RESPOND: Overview
- 3. RESPOND: Description
- 4. Cross reference for RESPOND classes, properties and dataproperties
  - 4.1. Classes
  - 4.2. Object Properties
  - 4.3. Data Properties
  - 4.4. Named Individuals
- 5. References
- 6. Acknowledgements

Figure 8: RESPOND Ontology's documentation available online.

For URL stability and manageability purposes, the W3C Permanent Identifier Community Group's[21] w3id.org[22] redirection service is used. The purpose of this initiative is to provide a secure, permanent URL re-direction service for Web applications.

W3C's Data on the Web Best Practices[23] states that providing metadata is a fundamental requirement that helps human users and computer applications to understand the data as well as other important aspects that describes a dataset. There are different guidelines available for describing ontology metadata, and the guideline described by Garijo and Poveda-Villallón [13] was followed as it was considered the most complete guideline among the ones reviewed.

This ontology documentation management was made manually during the first stages of the development phase. Afterwards, the OnToology[24] system developed by the OEG[25] was used. OnToology is a system to automate part of the collaborative ontology development process. Given a repository, OnToology will explore it to produce a complete ontology documentation (with WIDOCO) and enable leveraging the w3id.org redirection service.

[21] https://www.w3.org/community/perma-id/
[22] https://w3id.org/
[23] https://www.w3.org/TR/dwbp/
[24] http://ontoology.linkeddata.es/
[25] http://www.oeg-upm.net/

4.4 Ontology Instantiation

The RESPOND ontology's instantiation, often referred to as the ABox, was generated leveraging the pilot sites data point lists. These data point lists are captured in Excel sheets and contain information regarding the devices installed in pilot sites and include additional information such as their location and the property they measure. Figure 9 shows an excerpt of data point lists for the Madrid pilot site.

		Location				Appliance	
Device_id	**Gateway_id**	**Apartment_id**	**Location in the apartment**	**Location in the room**	**Device_type**	**Type**	**Brand**
ENE-11000133	ENE-526D97283FA69AC6	Madrid_05	living_room	other	sensor_humidity	other	N/A
ENE-11000133	ENE-526D97283FA69AC6	Madrid_05	living_room	other	sensor_co2	other	N/A
ENE-0C0005CA	ENE-526D97283FA69AC6	Madrid_05	living_room	other	meter_demand	AirConditioner	N/A
ENE-08000794	ENE-526D97283FA69AC6	Madrid_05	bedroom_1	other	sensor_temperature	other	N/A
ENE-0700034F	ENE-526D97283FA69AC6	Madrid_05	kitchen	other	meter_demand	Dishwasher	N/A

Figure 9: Madrid Data Point List excerpt.

Population of ontologies can be done in many different ways. As a matter of fact, there are different tools that support the automatization of this task. For example, [14] proposes the combination of LODRefine tool (currently known as OpenRefine[26]), SPIN[27] mappings and SPARQL Update[28] queries. In the RESPOND project, this ontology instantiation has been performed with a service based on Apache Jena framework[29]. Apache Jena is a free and open source Java framework for building Semantic Web and Linked Data applications. The developed service enables the extraction of the information from the Excel sheets, its semantic annotation with appropriate ontology terms, and its storage to an RDF Store where it will remain accessible via SPARQL queries.

Figure 10 shows the representation of a Develco Smart Plug that measures the electric consumption and controls the activation and deactivation of a dishwasher installed in the kitchen of a pilot house in Denmark. It is worth noting that the representation of measurements and actuations made by IoT systems and their storage in RDF graphs is not advisable, due to their poor performance. Instead, it is more suitable to store this IoT data in a time series database, which are optimized for time-series values. Therefore, the semantic representation of a device will contain the identifier of the device's measurements in the time series database. As shown in the image, *DEV-0015BC002F0002AB_demand_1* is the identifier used in InfluxDB to store the dishwasher's consumption data, and *DEV-0015BC002F0002AB_onoff_1* to store the activation state of the smart plug connected to the dishwasher.

[26] http://openrefine.org/
[27] http://www.w3.org/Submission/spin-overview/
[28] http://www.w3.org/TR/sparql11-update/
[29] https://jena.apache.org/

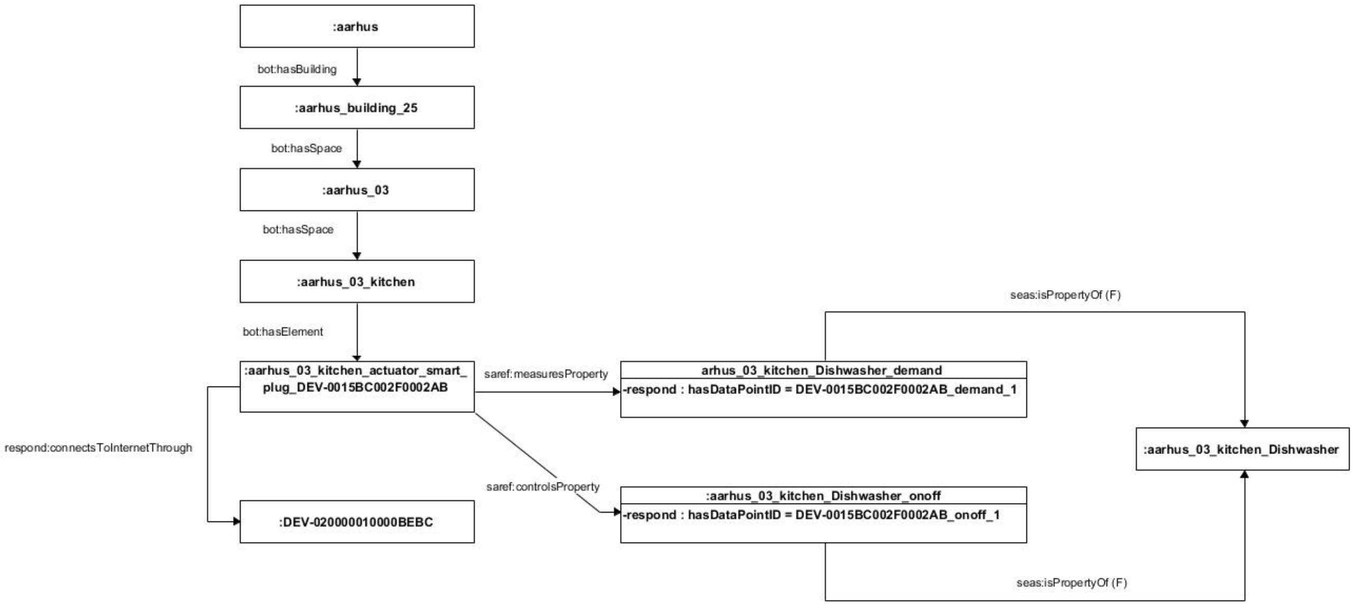


Figure 10: A Develco Smart Plug's representation using RESPOND ontology.

4.5 Ontology In Use

Thanks to this semantic representation, different information can be queried. For example, the list of devices installed within a house can be retrieved executing the SPARQL query shown in Listing 1, by replacing the wild card *$HOUSE_ID* is replaced by the target house's identifier.

```
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX bot: <https://w3id.org/bot#>

SELECT DISTINCT ?deviceID
WHERE{
        {?space bot:hasElement ?device;
                dc:identifier ?houseID.}
        UNION
        {?space bot:hasSpace ?subspace;
                dc:identifier ?houseID.
        ?subspace bot:hasElement ?device.}
        ?device dc:identifier ?deviceID.
  FILTER(
        ?houseID = $HOUSE_ID
  )
}
```

Listing 1: SPARQL query to retrieve the list of devices installed within a house.

Data collected by deployed meters, sensors or actuators deployed installed within houses can also be checked, such as the electric consumption of a house. This information, which is gathered by an electricity meter, is stored in InfluxDB. However, in order to query this information, it is necessary to know the Data

Point ID of the electricity meter at hand. To do so, the SPARQL query shown in Listing 2 is executed, where the wild card *$HOUSE_ID* is replaced by the corresponding house's ID.

```
PREFIX respond: <https://w3id.org/def/respond#>
PREFIX seas: <https://w3id.org/seas#>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX saref: <https://w3id.org/saref#>

SELECT ?dataPointID
WHERE{
        ?device saref:measuresProperty ?property;
            dc:identifier ?deviceID.
        ?property rdf:type respond:ElectricConsumption;
            seas:isPropertyOf ?foi;
             respond:hasDataPointID ?dataPointID.
        ?foi dc:identifier ?houseID.
  FILTER(
        ?houseID = $HOUSE_ID
  )
}
```

Listing 2: SPARQL query to retrieve the electric consumption of a house.

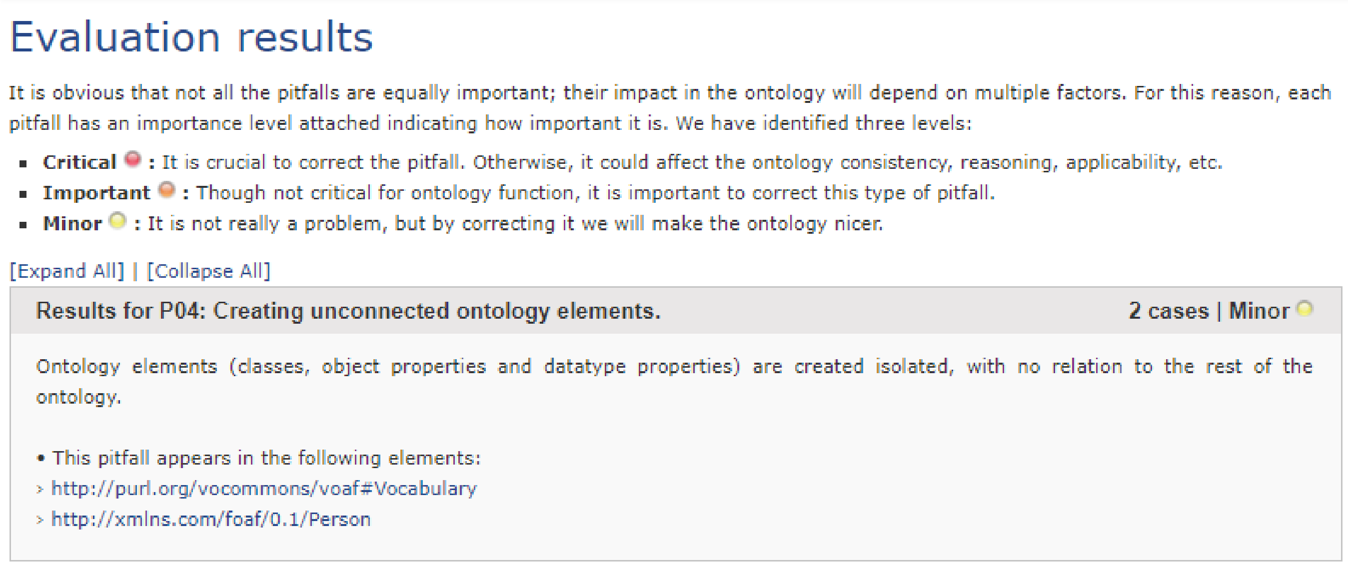
Furthermore, the semantic representation of this data is multilingual. Since this information is expected to be used by people from the three pilot sites (i.e. the Aran Islands, Aarhus and Madrid), some information is represented in English, Danish and Spanish. This multilingual feature is expected to enhance user-engagement and it can be exploited via SPARQL queries. The example shown in Listing 3 retrieves the description of a given device. The wildcard *$LANGUAGE* has to be replaced by "en" for English, "dk" for Danish and "es" for Spanish, and *$DEVICE_ID* with the corresponding device's serial number.

```
PREFIX respond: <https://w3id.org/def/respond#>
PREFIX dc: <http://purl.org/dc/terms/>

SELECT ?deviceName ?deviceDescription
WHERE{
  ?device dc:identifier ?devId;
          rdf:type ?deviceType.
  ?deviceType rdfs:label ?deviceName;
              rdfs:comment ?deviceDescription.

  FILTER (
    lang(?deviceDescription)= $LANGUAGE &&
    ?devId = $DEVICE_ID
  )
}
```

Listing 3: SPARQL query to retrieve the description of a device.

4.6 ONTOLOGY EVALUATION

The RESPOND ontology's design correctness was evaluated with OOPS! (OntOlogy Pitfall Scanner) [15], which detects some of the most common pitfalls appearing within ontology developments. OOPS! is available online[30] and evaluates an ontology against a catalogue of 41 potential pitfalls classified into three levels according to their severity: minor, important and critical. The use of this tool contributed to an early detection of pitfalls and complemented the manual review of the ontology's correctness. Results showed that only two minor issues were detected in the current ontology version. These were related to creating two unconnected ontology classes: *voaf:Vocabulary* and *foaf:*Person. These minor issues were reviewed and were considered as acceptable issues, since their impact with regards to the ontology's correctness is nearly inexistent. Figure 11 summarizes the pitfalls encountered in the RESPOND ontology.



Evaluation results

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- **Critical** : It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- **Important** : Though not critical for ontology function, it is important to correct this type of pitfall.
- **Minor** : It is not really a problem, but by correcting it we will make the ontology nicer.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements. **2 cases | Minor**

Ontology elements (classes, object properties and datatype properties) are created isolated, with no relation to the rest of the ontology.

• This pitfall appears in the following elements:
› http://purl.org/vocommons/voaf#Vocabulary
› http://xmlns.com/foaf/0.1/Person

Figure 11: RESPOND Ontology's evaluation with the OOPS! tool.

[30] http://oops.linkeddata.es/

5. Semantic Information Repository

Once all the required information is semantically annotated with appropriate ontology terms, it has to be stored in a suitable store where it will remain available. RDF Stores or triplestores are storage systems that are optimized for hosting this type of data and that, usually, support a SPARQL endpoint where data can be queried using SPARQL queries.

Currently, there are many different RDF store systems, and each of them has different capabilities. Therefore, deciding which of them is the most suitable for RESPOND required from an analysis. Figure 12 shows the DB-Engines[31] Ranking of RDF Stores as of March 2019.

☐ include secondary database models   19 systems in ranking, March 2019

Rank			DBMS	Database Model	Score		
Mar 2019	Feb 2019	Mar 2018			Mar 2019	Feb 2019	Mar 2018
1.	1.	1.	MarkLogic	Multi-model	13.74	-1.25	+2.77
2.	2.	↑ 3.	Virtuoso	Multi-model	3.20	+0.26	+1.37
3.	3.	↓ 2.	Apache Jena - TDB	RDF	1.94	-0.24	-0.35
4.	4.	4.	Amazon Neptune	Multi-model	1.03	-0.03	+0.33
5.	↑ 6.	↑ 7.	GraphDB	Multi-model	0.93	+0.07	+0.48
6.	↓ 5.	↓ 5.	AllegroGraph	Multi-model	0.88	-0.01	+0.24
7.	7.	↓ 6.	Stardog	Multi-model	0.77	+0.08	+0.24
8.	8.	↑ 12.	Blazegraph	Multi-model	0.55	+0.02	+0.40
9.	9.	9.	Redland	RDF	0.38	0.00	+0.12
10.	↑ 11.	↑ 15.	4store	RDF	0.35	+0.01	+0.24
11.	↓ 10.	↓ 10.	RDF4J	RDF	0.35	+0.00	+0.14
12.	12.	↓ 8.	Algebraix	RDF	0.27	+0.03	-0.09

Figure 12: DB-Engines Ranking of RDF Stores

Out of the identified first 12 RDF Stores in this ranking, RDF4J, Stardog and Virtuoso were the only ones with which RESPOND partners had previous experiences. Therefore, these three RDF Stores were compared with each other in more depth. Apart from previous experiences, articles such as [16] and [17] were also taken into account. Table 1 depicts the summary of the comparison made between these three RDF Stores.

	RDF4J	Stardog	Virtuoso
License	Open Source	Commercial (free limited Community edition)	Open Source
SPARQL endpoint	Yes	Yes	Yes

[31] https://db-engines.com/

Triple Capacity	?	10 Billion Triples	50 Billion Triples
Inference capability	No	OWL 2 + SWRL	Limited to some basic OWL reasoning
Access control	No	Access rights for users and roles	Fine grained access rights according to SQL-standard

Table 1: Comparison between RDF4J, Stardog and Virtuoso RDF Stores.

After this comparison, it was concluded that Stardog was the best option, mainly due to its inferencing capabilities. Providing inferred axioms at run-time rather than materialising them in the graph has the benefit that outdated or redundant information is avoided. Therefore, a Stardog version 6.0.1 was installed and made available online, where pilot site information was stored. However, due to the cease of Stardog's Community version after 31 March 2019, it was decided that a change of RDF Store was needed.

Since the remaining analysed two RDF Stores did not offer inferencing capabilities, the two of them were at the same position with respect to their selection. Finally, OpenLink Virtuoso was selected because there was a bigger experience comparing with RDF4J.

As to solve Virtuoso's lack of inference capabilities, RESPOND suggested a new SPARQL querying approach. Instead of directly querying the RDF Store with an SPARQL query, an intermediate service was created. This service, which is based on the Jena framework, loads the content of the RDF Store and executes a reasoning engine (e.g. Pellet reasoner[32]) to make inferences in existing data. This reasoning engine infers new implicit knowledge, and this is where the SPARQL query checks for the information. Therefore, the initial SPARQL query is executed over the set of knowledge that contains inferred data. It is important to note that inferences are not stored in the RDF Store and materialised in the graph, thus avoiding outdated or redundant information. Figure 13 represents the approach proposed by RESPOND.

[32] https://www.w3.org/2001/sw/wiki/Pellet

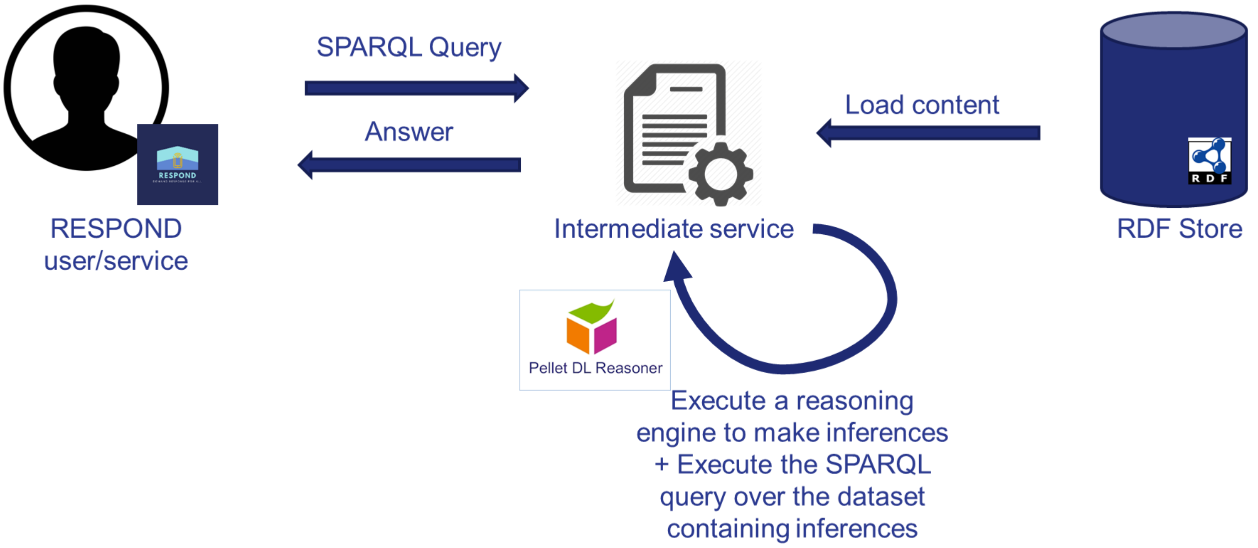


Figure 13: RESPOND's inferencing approach.

In this case, the proposed approach has been applied to a Virtuoso RDF Store. Due to the good software engineering practices applied, this same approach could be easily extended for other RDF Stores.

The inferencing capabilities are essential to enable the discovery of relationships that are not explicitly stated on the semantic repository, which has a twofold benefit. On the one hand, it allows simplifying the annotation of the data to be stored on the semantic repository, and on the other, it eases keeping the semantic data updated.

This inferencing capability enables exploiting both OWL type rules (e.g. property chains) as well as rules defined using other languages. For instance, the OWL axioms defined within the BOT ontology, defines an object property *bot:hasElement* which links a zone to an element that it is contained in. This object property is defined as sub property chain of *bot:containsZone o bot:hasElement*. As a matter of fact, the intended use of *bot:hasElement* is not to be stated explicitly, but to be inferred from its sub-properties. It will, for example, allow one to query for all the doors of a building given that they have an adjacency to spaces of the building.

This is just an example of how enabling the inferencing capabilities allows the discovery of relevant information for the RESPOND solution.

5.1 Ontology as Integration Layer

Following an extensive overview of the main ontology aspects, existing ontologies that REPSOND will leverage upon and guidelines for its implementation, the following description is primarily focused on the elaboration of the potential benefits from having an ontology-based metadata layer as a common integration and interoperability platform for diverse, heterogeneous ICT systems.

Namely, the proposed metadata layer, leveraging upon the ontology-based knowledge store and Canonical Data Model (CDM) as uniform messaging format, aims primarily at providing semantic enrichment of signals coming from BMS, additional sub-metering and data loggers, or external sources, thus offering critical contextual information to the high-level energy management applications.

Furthermore, the system components will be able to share the knowledge about the physical system and will understand each other when referring to the same concepts. For instance, the underlying supervision and control platform (being either BMS, EMS or IoT-based monitoring platform) typically uses its proprietary data naming format for the representation of the sensor readings, whereas, an external module/application may use its own data naming format. In this way, the high-level energy management applications, communicating the acquired data and proposed actions to the end user, can provide semantically enriched information about particular energy asset, such as its technical properties, interconnection with other assets, physical location within the infrastructure etc., instead of the non-intuitive asset identifiers (IDs). All this heterogeneity in different systems and data formats introduces critical interoperability issues and requires tedious translation procedures in communication between individual system components. The proposed metadata layer aims to solve this problem by leveraging its implementation on the utilization of a holistic ontology-based facility data model. Thus, the three fundamental features are enabled:

1. the knowledge about the system is centralized and stored within the ontology-based facility data model, providing critical semantic relations within the domain of interest;
2. by establishing a common vocabulary of the system entities, the facility data model enables all system components to “understand” each other when referring to a particular entity, in a seamless and transparent way;
3. having a centralized knowledge repository, all relevant information is updated, consistent, synchronized and accurate.

Considering the proposed metadata layer is based on the ontology, its integration into the overall RESPOND solution is provided by the corresponding knowledge discovery service, which involves specific ontology interface. The main responsibility of such interface is to enable knowledge extraction, as well as the delivery of the requested information, such as technical characteristics of the field-level energy assets, supervision and control of devices and systems, their topological information etc., in a transparent way. In other words, a custom designed application program interface (API) is necessary to integrate ontology-based knowledge repository and provide seamless extraction, acquisition and reasoning for the rest of system components. In relation to this, a detailed ontology-supported information flow and knowledge extraction is depicted in Figure 14.

As previously mentioned, the relevant information for the RESPOND platform energy management applications come either from a resource layer, comprising BMS system, meters etc., or various external sources, comprising meteorological forecast service or energy pricing forecast service. Furthermore, additional measurements such as indoor ambient, or people presence, may come from an independent IoT based monitoring/sensing platform and could be used to enrich diagnostics of the most critical energy consuming devices (e.g. heating is used in room with no people, or there is an open window). In either way, the relevant information is first wrapped into a custom CDM (XML/JSON format) at the corresponding gateway according to a custom, pre-defined, data naming convention. The data are then forwarded to the middleware layer. Collected data, carried within the CDM formatted messages, may contain a wide range of performance monitoring parameters including electricity consumption measurements, on-site production, storage status etc. However, the CDM messages, gathered within the middleware layer, do not carry the semantics related to the data point they represent.

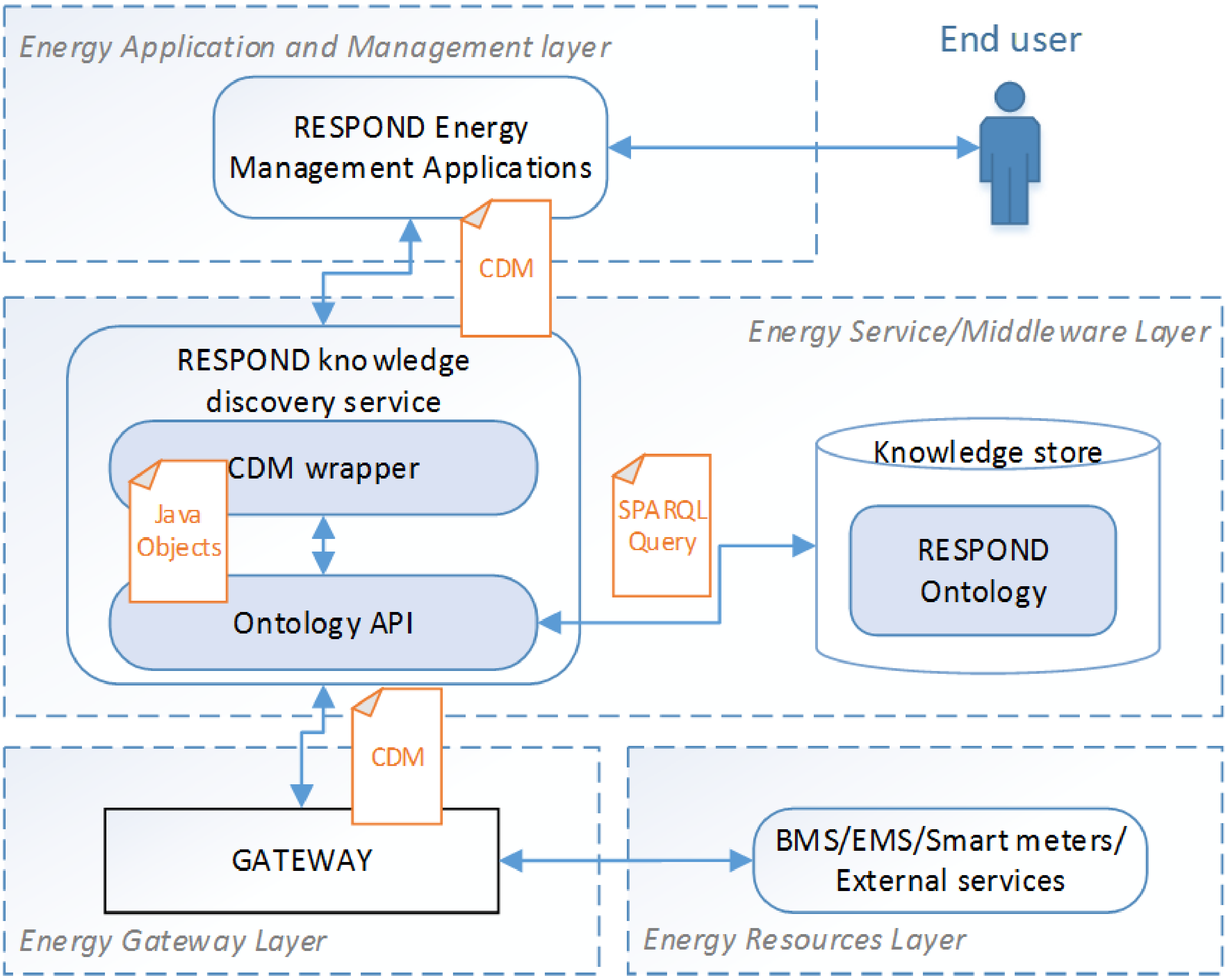


Figure 14: Information flow and knowledge extraction

To illustrate, a data point is labelled by a specific system defined ID, which may, in general, even represent a random set of characters, having no association with the device it represents nor its attributes (like the rated power or specific deployment location), whereas the entire semantics and contextual knowledge is stored within the ontology. Otherwise, the middleware would have to be aware of, and exchange custom messages with each information source, following their proprietary data naming convention, in order to properly interpret the information of interest.

This issue is, therefore, bridged with the introduction of a common metadata layer, hosting the facility data model and accompanying API. Each CDM message is first parsed in order to acquire the source device ID (for instance, device that triggered an alarm) and corresponding measurement/status value. The extracted device ID is then used to query the ontology in order to obtain additional relevant information, such as full specification of the device, which system/sub-system it belongs to, what the neighbouring devices are, where it is located within the facility itself.

Querying is performed through the ontology API, which has several functionalities ranging from generation of SPARQL queries to communication with the ontology data store and offering the gathered information in appropriate format to the rest of the system. The facility ontology can reside, without any restrictions, either locally, within the RESPOND's deployment environment, or on a remote server, serving as a remote knowledge store (e.g. using Virtuoso Universal Server). Regardless of the implementation, the overall communication chain is completely transparent for the stakeholders, owing to the developed API. In this way, the facility ontology model can provide critical semantic enrichment of the signals coming from devices/systems, thus enabling the energy management applications to deliver high-level, immediately applicable, energy conservation measures to end-users.

5.2 Scenarios for Semantic Information Exploitation

The following are several examples, illustrating the employment of the relevant ontology API functionalities:

- *Technical characterization and semantic interpretation of signals* – enables field-level asset/device semantic characterization based on the unique device ID, embedded within the messages/signals coming from remote terminal units/devices/sensors, providing relevant technical parameters (extracted via SPARQL queries) such as device rated power, efficiency or capacity or topology information such as to which system it belongs, how and to which device/component it is connected, where it is located etc.
- *Update of facility ontology model* – enables the update of specific class instances and their properties, based on the received information/data from sensors/systems (using SPARQL Update commands). Update arguments are considered to be both unique device identifier and desired property value. This scenario is essential to keep the model updated whenever pilot site devices are changed, or even new pilot houses are incorporated into the project.
- *Application of a generic inference engine* – in order to maintain consistency of the data stored in the ontology, a set of rules defined whether in OWL or in the Semantic Web Rule Language (SWRL) is used by the suitable inference engine, which is responsible for reasoning upon the class instances. For instance, corresponding relations among class instances could be automatically established as a

result of reasoning, such as in the case of transitive relations (e.g. if there is a device belonging to a specific system, then the components of this device belong to the same system as well). These inferencing capabilities enables the discovery of relationships that are not stated explicitly, thus simplifying the annotation of data.

6. CONCLUSIONS

One of the main outputs of this deliverable is the RESPOND Ontology, which is based on the NeOn Methodology and following the best practices of reusing existing well-known ontologies. This reuse is not a trivial task, due to the excess of existing ontologies nowadays and their different level of coverage. Therefore, a clear specification of ontology requirements is necessary and a thorough analysis of ontologies.

As the ontology requirements change, these reused ontologies may need to be extended with additional axioms. Furthermore, with these extensions new problems may arise. In this regard, basing the ontology development on well-defined methodologies is essential in order to ensure having available mechanisms for dealing with these new problems.

The representation of raw data with appropriate ontological resources ensures the enhanced performance of different services. For example, analytical services such as data-driven predictive models may produce more accurate outcomes. Furthermore, taking leverage of this semantically annotated data also targets at ensuring the coherency of the results produced by different services that rely on data.

Last but not least, capturing domain and expert knowledge in ontologies enables the inference of new knowledge from implicit axioms. This capability is key for opening up new opportunities such as the development of new or enhanced functionalities in existing services. The inferencing of new data is not at hand with the use of traditional technologies, and the RESPOND solution develops an approach to ensure this capability, being independent from the RDF Store system used.

REFERENCES

[1] T. Heath and C. Bizer, Linked data: Evolving the web into a global data space, Synthesis lectures on the semantic web: theory and technology, Vol. 1, Morgan & Claypool Publishers, 2011, pp. 1-136. https://doi.org/10.2200/S00334ED1V01Y201102WBE001 .

[2] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez and A. Arnaiz, EEPSA as a core ontology for energy efficiency and thermal comfort in buildings, Semantic Web Journal (Under review)."

[3] P. Pauwels and W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, Automation in Construction 63 (2016), 100-133. https://doi.org/10.1016/j.autcon.2015.12.003 .

[4] D. Bonino and F. Corno, Dogont - Ontology Modeling for Intelligent Domotic Environments, in: International Semantic Web Conference, Springer, 2008, pp. 790-803. https://doi.org/10.1007/978-3-540-88564-1_51 .

[5] M.H. Rasmussen, P. Pauwels, C.A. Hviid and J. Karlshoj, Proposing a Central AEC Ontology That Allows for Domain Specific Extensions, in: Joint Conference on Computing in Construction, Vol. 1, 2017, pp. 237-244. https://doi.org/10.24928/JC3- 2017/0153 .

[6] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson and A. Herzog, The SSN ontology of the W3C semantic sensor network incubator group, Web Semantics: Science, Services and Agents on the World Wide Web 17 (2012), 25-32. https://doi.org/10.1016/j.websem.2012.05.003 .

[7] A. Haller, K. Janowicz, S. Cox, M. Lefrançois, K. Taylor, D.L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson and C. Stadler, The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation, Semantic Web To be published (2018). https://doi.org/10.3233/SW-180320 .

[8] L. Daniele, F. den Hartog and J. Roes, Created in close interaction with the industry: the smart appliances reference (SAREF) ontology, in: International Workshop Formal Ontologies Meet Industries, Springer, 2015, pp. 100-112. https://doi.org/10.1007/978-3-319-21545-7_9 .

[9] M. Lefrançois, Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns, in: Proceedings of Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT, 2017.

[10] A. Gyrard, A. Zimmermann and A. Sheth, Building IoT based applications for Smart Cities: How can ontology catalogs help?, IEEE Internet of Things Journal (2018).

[11] A. Lozano-Tello and A. Gómez-Pérez, Ontometric: A method to choose the appropriate ontology, Journal of Database Management (JDM) 15(2) (2004), 1-18.

[12] D. Garijo, WIDOCO: A Wizard for Documenting Ontologies, in: The Semantic Web - ISWC 2017, C. d'Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange and J. Hein, eds, Springer International Publishing, Cham, 2017, pp. 94{102. ISBN 978-3-319-68204-4.

[13] D. Garijo and M. Poveda-Villalón, A checklist for complete vocabulary metadata, Technical Report, 2017. https://w3id.org/widoco/bestPractices.

[14] N. Tomasevic, M. Batic, L.M. Blanes, M.M. Keane, S. Vranes : Ontology based facility data model for energy management. Adv. Eng. Inform. 29(4), 971-984 (2015). https://doi.org/10.1016/j.aei.2015.09.003

[15] M. Poveda-Villalón, M.C. Suárez-Figueroa, & A. Gómez-Pérez. (2012, October). Validating ontologies with oops!. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 267-281). Springer, Berlin, Heidelberg.

[16] G. A. Atemezing, F. Amardeilh. (2018, June). Benchmarking commercial RDF stores with publications office dataset. In *European Semantic Web Conference* (pp. 379-394). Springer, Cham.

[17] P. Bellini, P. Nesi. (2018). Performance assessment of RDF graph databases for smart city services. *Journal of Visual Languages & Computing*, *45*, 24-38.