Project Acronym: **BIMERR**
Project Full Title: **BIM-based holistic tools for Energy-driven Renovation of existing Residences**
Grant Agreement: **820621**
Project Duration: **45 months**

DELIVERABLE D4.3
BIMERR Ontology & Data Model 2

Deliverable Status: **Final**
File Name: **D4.3-v0.0.9.docx**
Due Date: **30/06/2021 (M30)**
Submission Date: **30/06/2021 (M30)**
Task Leader: **UPM (T4.2)**

Dissemination level	
Public	X
Confidential, only for members of the Consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621

The BIMERR project consortium is composed of:		
FIT	Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V.	Germany
CERTH	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis	Greece
UPM	Universidad Politécnica De Madrid	Spain
UBITECH	Ubitech Limited	Cyprus
SUITE5	Suite5 Data Intelligence Solutions Limited	Cyprus
HYPERTECH	Hypertech (Chaipertek) Anonymos Viomichaniki Emporiki Etaireia Pliroforikis Kai Neon Technologion	Greece
MERIT	Merit Consulting House Sprl	Belgium
XYLEM	Xylem Science And Technology Management Gmbh	Austria
CONKAT	Anonymos Etaireia Kataskevon Technikon Ergon, Emporikon Viomichanikonkai Nautiliakon Epicheiriseon Kon'kat	Greece
BOC	Boc Asset Management Gmbh	Austria
BX	Budimex Sa	Poland
UOP	University Of Peloponnese	Greece
UEDIN	University of Edinburgh	United Kingdom
UCL	University College London	United Kingdom
NT	Novitech As	Slovakia
FER	Ferrovial Agroman S.A	Spain

Disclaimer

BIMERR project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°820621. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Commission (EC). EC is not liable for any use that may be made of the information contained therein.

AUTHORS LIST

Leading Author (Editor)				
	Surname	First Name	Beneficiary	Contact email
	Chávez-Feria	Serge	UPM	serge.chavez.feria@upm.es
	Poveda-Villalón	Maria	UPM	mpoveda@fi.upm.es
Co-authors (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Bountouni	Nefeli	Suite5	nefeli@suite5.eu
2	Fenz	Stefan	Xylem	fenz@xylem-technologies.com
3	Bosché	Frédéric	UEDIN	f.bosche@ed.ac.uk
4	Giannakis	Giorgos	Hypertech	g.giannakis@hypertech.gr
5	González-Gerpe	Salvador	UPM	sgonzalez@delicias.dia.fi.upm.es
6	Katsos	Marios	Hypertech	m.katsos@hypertech.gr
7	Kousouris	Spiros	Suite5	spiros@suite5.eu
8	Thanos	Tsakiris	CERTH	atsakir@iti.gr
9	Lampathaki	Fenareti	Suite5	fenareti@suite5.eu
10	Valero	Enrique	UEDIN	e.valero@ed.ac.uk
11	Katsigarakis	Kyriakos	UCL	k.katsigarakis@ucl.ac.uk
12	Kompos	Kostas	Hypertech	k.kompos@hypertech.gr
13	Falcioni	Damiano	BOC	damiano.falcioni@boc-eu.com
14	Petridis	Kosmas	Hypertech	k.petridis@hypertech.gr
15	Straka	Martin	NT	straka@novitechgroup.sk
16	Varga	Jan	NT	varga@novitechgroup.sk
17	Lilis	Georgios	UCL	gnl2@cornell.edu

REVIEWERS LIST

List of Reviewers (in alphabetic order)				
#	Surname	First Name	Beneficiary	Contact email
1	Giannakis	Giorgos	Hypertech	g.giannakis@hypertech.gr
2	Katsigarakis	Kyriakos	UCL	k.katsigarakis@ucl.ac.uk

REVISION CONTROL

Version	Author	Date	Status
0.1	UPM	01-04-2021	Added ToC
0.2	UPM	31-05-2021	First version section 1 and 2
0.3	UPM, Suite5, Xylem, UEDIN, Hypertech, CERTH, UCL, BOC, NT	04-06-2021	Added section 3
0.4	UPM	15-06-2021	Overall content review
0.5	UPM	15-06-2021	Overall format review
0.6	UPM	15-06-2021	First version
0.7	UPM, Hypertech, UCL	25-06-2021	Internal review corrections
0.8	UPM	28-06-2021	First version
0.9	UPM	30-06-2021	Coordinator review and changes

TABLE OF CONTENTS

List of Figures

List of Tables

ACRONYMS

Acronym	Meaning
BIM	Building Information Modelling
BICA	Building Information Collection Application
BCF	Building Collaboration Format
BIF	BIMERR Interoperability Framework
BO2DM	BIMERR Ontology to Data Model
HVAC	Heating, Ventilating and Air Condition
IFC	Industry Foundation Classes
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LOT	Linked Open Terms
ORSD	Ontology Requirement Specification Document
OWL	Web Ontology Language
obXML	Occupant Behaviour XML
PWMA	Project Workflow Management
PRUBS	Profile Resident Usage of Building System
RenoDSS	Renovation Decision Support System
RDF	Resource Description Framework
SenML	Sensor Measurement List
TTL	Terse RDF Triple Language
UML	Unified Modelling Language
URI	Uniform Resource Identifier
WGS	World Geodetic System
XSD	XML Schema Definition
XML	eXtensible Markup Language

EXECUTIVE SUMMARY

The objective of this deliverable is to report on the final status of the ontology development process started in D4.2. For that purpose, the document starts by giving a general overview of the current state of the BIMERR ontology network at M30, and the resources generated during its design, implementation, and publication.

Each module of the BIMERR ontology developed in this second phase is described in detail, indicating its requirements and the current model description. Domains that are covered in this document include: Building, Sensor Data, Weather, Key Performance Indicator, Occupancy Profile, Annotation Objects, Information Objects, Renovation Processes, and Material Properties. The whole suite of ontologies is available online in different formats on the BIMERR ontology network portal.[1]

The document also covers the alignment and translation from the BIMERR ontology to a JSON serialized data model. First, the metadata ontology is explained, providing a description of the terms included in the model to allow the enrichment of the ontologies. Afterwards, the transformation process is summarized showing the resulting data model.

Finally, a summary of the methodology followed for the creation of the models is described, being mostly focused on the particular modifications made to the original methodology described in D4.2 in order to adapt it to the complexity of the project and the number of domains.

[1] https://bimerr.iot.linkeddata.es/

1 INTRODUCTION

1.1 SCOPE AND OBJECTIVES OF THE DELIVERABLE

As explained in D4.2 “BIMERR Ontology & Data Model 1” [Chávez-Feria et. al., 2020], the BIMERR interoperability approach relies on a common data model and an ontology to be shared and understood by all components connected to the BIMERRR Interoperability Framework (BIF). More precisely, the common data model, as a JSON version, is derived from the BIMERR ontology. Information technologies took the term “ontology” from philosophy and redefined it as “formal, explicit specifications of a shared conceptualization [Studer et. a., 1998]. Concerning the BIMERR ontologies, they will be formalized following Description Logics and implemented in the W3C Web Ontology Language standard OWL.[2] As there is a broad number of domains to be considered in the BIMERR conceptualization, the ontology to be developed will be organized as a modular ontology network. Some modules, or parts of them, will be aligned with existing standards or well-known ontologies that will be reused whenever possible. It is worth reminding that a number of models and ontologies to be reused were already analyzed in D4.1 [Tiwari et.al. 2019].

The goal of this deliverable is to describe (a) the BIMERR ontology network and the resources generated during its development including a detailed description of each module, and (b) the transformation process to achieve the desired JSON data model, considering all the requirements for updates and changes that have been considered up to M30. Finally, this document presents some conclusions about the steps followed and proposes next steps.

Figure 1 represents the agile methodology followed during the ontology development process, described in detail in D4.2 [Chávez-Feria et. al., 2020]. This methodology is based on the LOT methodology[3] [Poveda-Villalón et. al., 2019b] which was originally proposed in

[2] https://www.w3.org/TR/owl-ref/
[3] https://lot.linkeddata.es/

[Poveda-Villalón, 2012] and [García-Castro et. al., 2017]. It is worth mentioning that some parts of the methodology have been adapted or specialized for the BIMERR project needs.

For the sake of understandability, readers not familiar with the concept of ontologies in computer science and the web are referred to "Ontology Development 101: A Guide to Creating Your First Ontology"[4] as a basic reference for this topic.

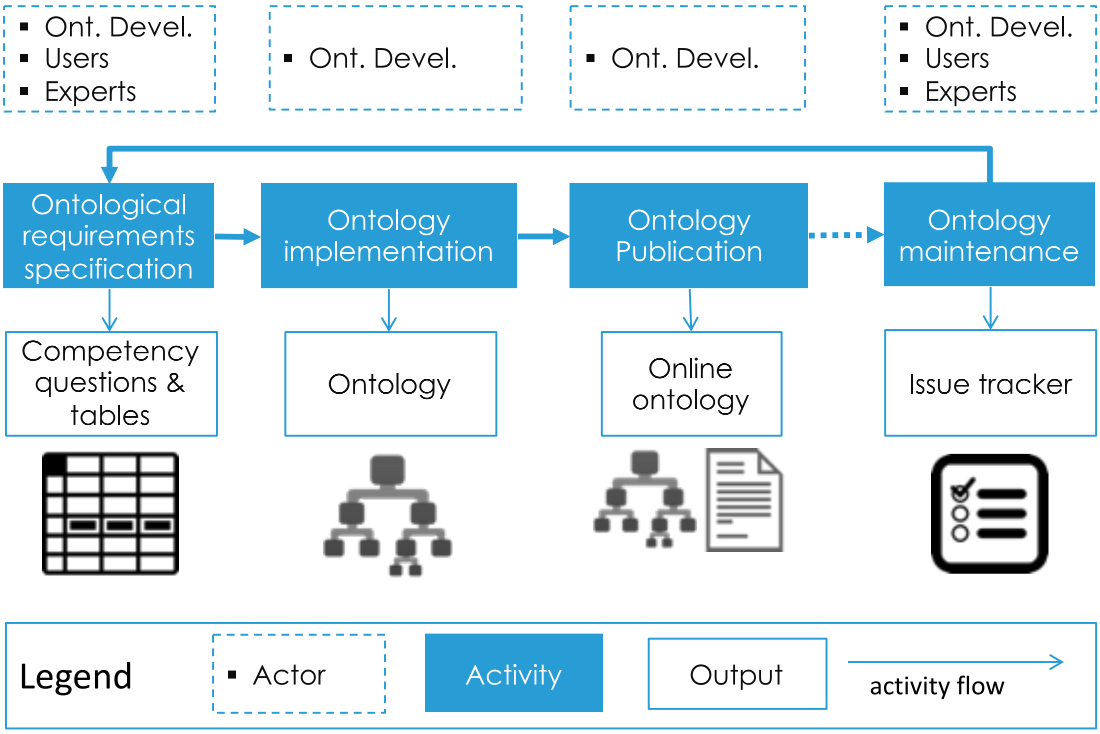


Figure 1. General overview of the ontology development process.

[4] http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

1.2 Relation to Other Tasks and Deliverables

This deliverable D4.3 is related to other BIMERR deliverables as indicated in Table 1.

Table 1. Relations to other deliverables

Deliverable	Deliverable Title	Relations and Contributions
D3.1	Stakeholder requirements for the BIMERR system.	Provides an initial set of high-level requirements for the development of the BIMERR ontology network.
D3.2	Survey of data models, ontologies and standards in the wider Energy Efficient Buildings domain.	The BIMERR ontology network is built upon well-known standards and existing ontologies that are described in this deliverable.
D4.2	BIMERR Ontology & Data Model 1	Describes the first version of the ontology network, and the technological infrastructure and tools that support its development.
D4.5	BIMERR Building Semantic Modelling tool 2	The final version of the Semantic Modelling Tool includes the Ontology Manager Framework component, which is in charge of storing, managing, and publishing the ontologies of the project.
D4.7	BIMERR Information Collection & Enrichment Tool 2	The final version of the Information Collection & Enrichment Tool includes the Knowledge Graph Generator component, which transform materials and weather data to RDF. The ontology network support this transformation providing the desired schema.
D5.2	Prototype of Enhanced BIM Platform 2	The BIMERR data model supports the exchange of enriched IFC data, provided by the BIM-MP, through the building and materials models.
D5.8	Building Resident Energy Related Behaviour Profiling Framework 2	The BIMERR data model supports the exchange of resident energy profiles, generated by PRUBS, through the Occupancy profile model.
D6.5	Renovation Process Simulation Tool 2	The BIMERR data model supports the exchange of KPI and processes data, provided by the PWMA tool, through the Key Performance Indicator and Renovation Processes models.
D6.7	Adaptive Workflow Management & Automation Tool 2	The BIMERR data model supports the exchange of work orders, provided by the PWMA tool, through the Renovation Processes model.
D6.8	Smart Glass Application for On-site Renovation Worker Support 1	The BIMERR data model supports the exchange of annotations regarding on-site related issues, provided by the ARIBFA tool, through the Annotation Objects model.
D6.11	Renovation Progress Monitoring & Alerting Application for Residents 2	The BIMERR data model supports the exchange of annotations and occupancy data, generated by BICA, through the Annotation Objects and Occupancy Profile models.
D7.10	Integrated BIMERR Renovation Decision Support System 2	The BIMERR data model supports the exchange of renovation measures, generated by RenoDSS, through the Key Performance Indicators model.

2 BIMERR Ontology Network and Data Models Portal

This section describes the whole ontology network developed for the BIMERR project as well as the main artefacts and resources obtained during its development. The models that have been created in this second iteration are analyzed in Section 3. Although this deliverable is focused on the design and implementation of the ontologies, it is important to remark that the results of this effort are both the ontologies and the data models that are finally supported by BIF.

Table 2 summarizes the status of the ontologies on the first and the second releases. During the development of the first sprint of the ontologies, the BIMERR network was devised to contain 8 modules. Two of those modules were: the Health & Safety Issues and the Renovation Measures modules. However, in this second sprint, it was realized that the concepts and relations inside both modules could be covered with other related ontologies, such as the Annotation Objects and the KPI modules, respectively. In addition to that, new domains were identified, more concretely the Renovation Processes and the Information Objects domains.

Table 2. Summary of ontology status.

Domain	First iteration	Second Iteration	Comments
Occupancy profile	First version	0.1.0	The new version was the product of new requirements that appear during the integration activities with BIF.
Weather	First version	0.0.1	-
Sensor data	First version	0.0.3	New version product of minor errors in the implementation.
Key performance indicators	First version	0.1.5	The new version was the product of new requirements from the RenoDSS tool.
Building	Draft	0.1.5	-
Material properties	Draft	0.1.0	-
Annotation objects	-	0.0.2	-
Renovation processes	-	0.0.1	-
Information objects	-	0.0.3	-

2.1 THE BIMERR ONTOLOGY NETWORK

The BIMERR ontology network consists of nine modules, where each module represents a domain of the project. Figure 2 provides a graphical overview of the network, showing the main concepts of each module and the relations between them. Each color is associated to a different domain in the network to ease the identification of models. Table 3 indicates the list of the prefixes and namespaces used to identify each domain along the document.

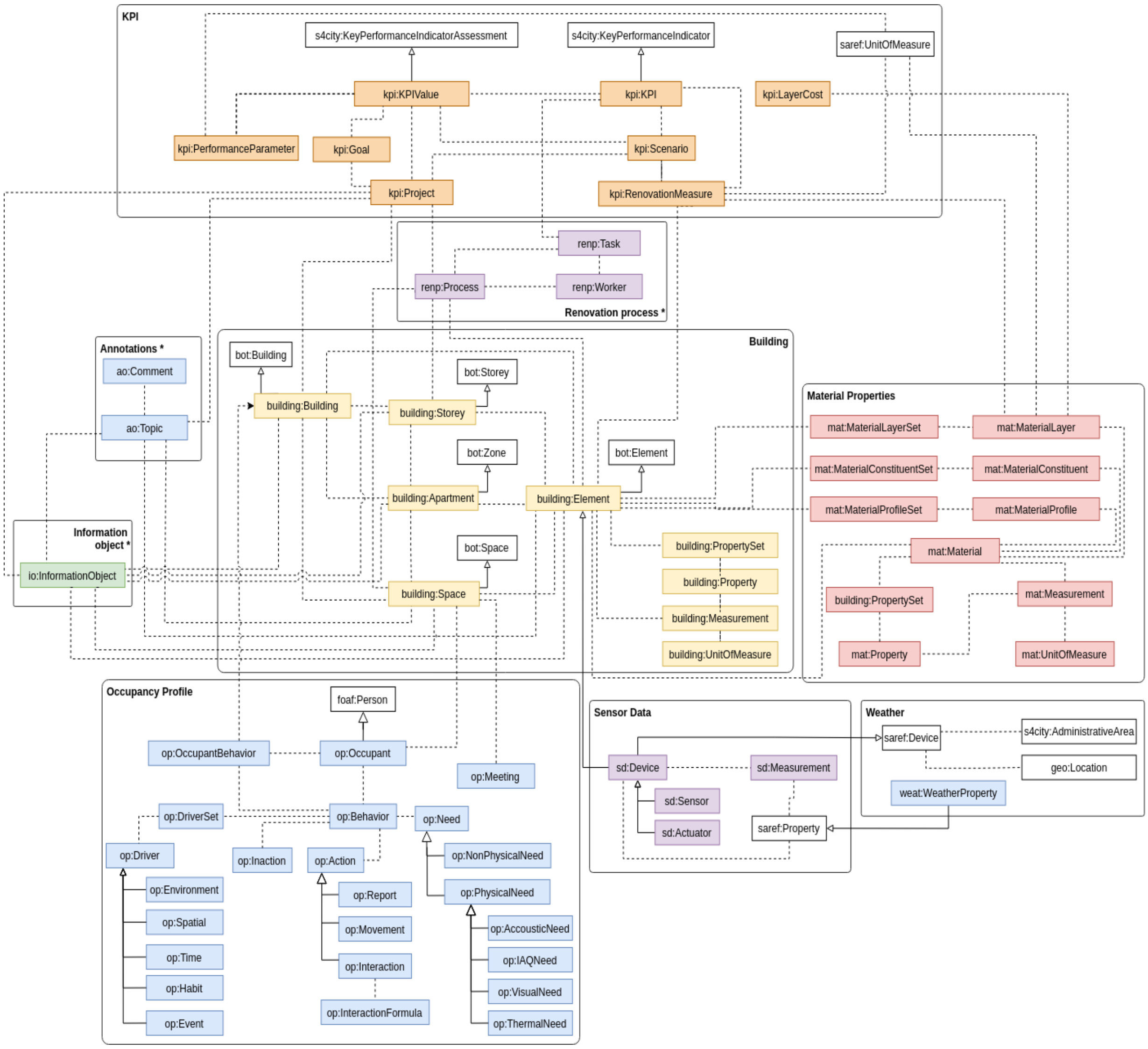


Figure 2. BIMERR ontology network overview

In addition, white boxes in Figure 2 represent reused terms from existing ontologies. For instance, in the building module, the concept `bot:Building` is taken from the BOT ontology, which is not a BIMERR ontology but a reused one. The list of prefixes, and corresponding ontologies, reused along the development of the BIMERR model are listed in Table 4. As it can be observed, some of the ontologies reviewed in previous steps of the project were reused, like SAREF[5] and geo,[6] as they are related to the project domain. Furthermore, additional cross-domain ontologies were reused once the requirements and conceptualizations were analyzed in detail, as the case of the SKOS (Simple Knowledge Organization System) and Time ontologies.

Table 3. List of published BIMERR ontologies and their corresponding namespaces

Prefix	Ontology namespace
building	https://bimerr.iot.linkeddata.es/def/building#
kpi	https://bimerr.iot.linkeddata.es/def/key-performance-indicator#
io	https://bimerr.iot.linkeddata.es/def/information-objects#
mat	https://bimerr.iot.linkeddata.es/def/material-properties#
ao	https://bimerr.iot.linkeddata.es/def/annotation-objects#
renp	https://bimerr.iot.linkeddata.es/def/renovation-processes#
op	https://bimerr.iot.linkeddata.es/def/occupancy-profile#
sd	https://bimerr.iot.linkeddata.es/def/sensor-data#
weat	https://bimerr.iot.linkeddata.es/def/weather#

As we can infer from Figure 2, an intricate relationship between the modules of the network exists, where the building ontology acts as the main bridge between the rest of the modules. These interconnections allow the integration of heterogeneous data, generated from the tools that compose the project, thus enabling the execution of complex queries that require data from different domains. The relationships between the modules are represented as arrows between the concepts of the ontology network, which formally are the equivalent to object properties in the ontology. For instance, the `building:Element` concept in the Building ontology is connected to the

[5] https://saref.etsi.org/core/v3.1.1/

[6] https://www.w3.org/2005/Incubator/geo/XGR-geo-ont-20071023/

`mat:MaterialLayerSey` concept in the Material Properties ontology through the relationship `mat:hasMaterialLayerSet`.

Table 4. List of reused ontologies and their corresponding namespaces

Prefix	Ontology namespace
foaf	http://xmlns.com/foaf/0.1/
geo	http://www.w3.org/2003/01/geo/wgs84_pos#
org	http://www.w3.org/ns/org#
xsd	http://www.w3.org/2001/XMLSchema#
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
saref	https://w3id.org/saref#
s4bldg	https://w3id.org/def/saref4bldg#
s4city	https://w3id.org/def/saref4city#
bot	http://bimerr.iot.linkeddata.es/def/building#
skos	http://www.w3.org/2004/02/skos/core#
time	http://www.w3.org/2006/time#

The main hierarchies between concepts are also included. These hierarchies are represented by arrows with white endings (triangles) and can be read as follows: The class in the origin of the arrow is a subclass of the class in the end of the arrow.

It is worth noting that complete and up to date documentation of each ontology module is provided online and is accessible through each of the ontologies' URI. In the rest of this document only the main concepts or modeling decisions are detailed.

Apart from the domain-specific ontological requirements, the BIMERR ontologies development is based on the following non-functional requirements:

- Reuse: existing ontologies or standard models shall be reused when possible, thereby increasing interoperability with external systems that might be already using such ontologies. The point is also applied at the meta-level by using standard technologies to implement the ontologies themselves.

- Modularity: the ontology shall be designed as a network in which modules might be interconnected and refer to others.
- Good practices: the ontologies shall be developed following methodologies and best practices commonly used in ontology engineering in order to address ontology development activities such as design, implementation, evaluation, publication, and documentation, among others.

The graphical representation of the ontology network as well as the diagrams that represent each module follow a particular visual notation that is also used along this document. The visual notation used in the project is the one formulated in the Chowlk framework, which defines a set of graphical shapes for the design of ontology conceptualizations. The use of this notation not only allows the creations of UML-like models that can be used to document the ontologies, but also the automatic generation of the ontology implementation in a formal language (OWL) by means of a converter that is part of the framework.[7] The complete documentation with respect to the notation can be found in the official website of the project.[8] Here we give a general overview of the conventions used in the diagrams, which is also detailed in Figure 3:

- Colored rectangles are used to denote classes created in the ontology being described, while white rectangles denote reused classes. For all the entities, it is indicated in which ontology they are defined by the prefix included before their identifier.
- Arrows are used to represent properties between classes and to represent some rdf, rdfs and owl constructs, more precisely:
    - Plain arrows with white triangles represent the `rdfs:subClassOf` relation between two classes. The origin of the arrows is the class to be declared as subclass of the class at the destination of the arrow.
    - Plain arrows between two classes indicate that the object property has declared as domain the class in the origin and as range the class in the

[7] https://chowlk.linkeddata.es/chowlk
[8] https://chowlk.linkeddata.es/chowlk_spec

destination of the arrow. The identifier of the object property is indicated within the arrow.

- Dashed labeled arrows between two classes indicate that the object property can be instantiated between the classes in the origin and the destination of the arrow. The identifier of the object property is indicated within the arrow.
- Dashed arrows with the identifiers between stereotype signs (i.e., “<< >>”) refer to OWL[9] constructs that are applied to some ontology elements, that is, they can be applied to classes or properties depending on the OWL construct being used.
- Dashed arrows with no identifier are used to represent the `rdf:type` relation, indicating that the element in the origin is an instance of the class in the destination of the arrow.

- Datatype properties are denoted by rectangles attached to the classes, in a UML-oriented way. Dashed boxes represent datatype properties that can be applied to the class it is attached to while plain boxes represent that the domain of the datatype property is declared to be the class attached.
- Individuals are denoted by rectangles in which the identifier is underlined. Literals are denoted by rectangles in which the value is included between quotation marks.
- The representation of additional property axioms (functional, inverse functional, transitive, and symmetric) that are being used in the diagram are shown in the overview ontology legend.

[9] https://www.w3.org/TR/owl-ref/#DatatypeProperty-def

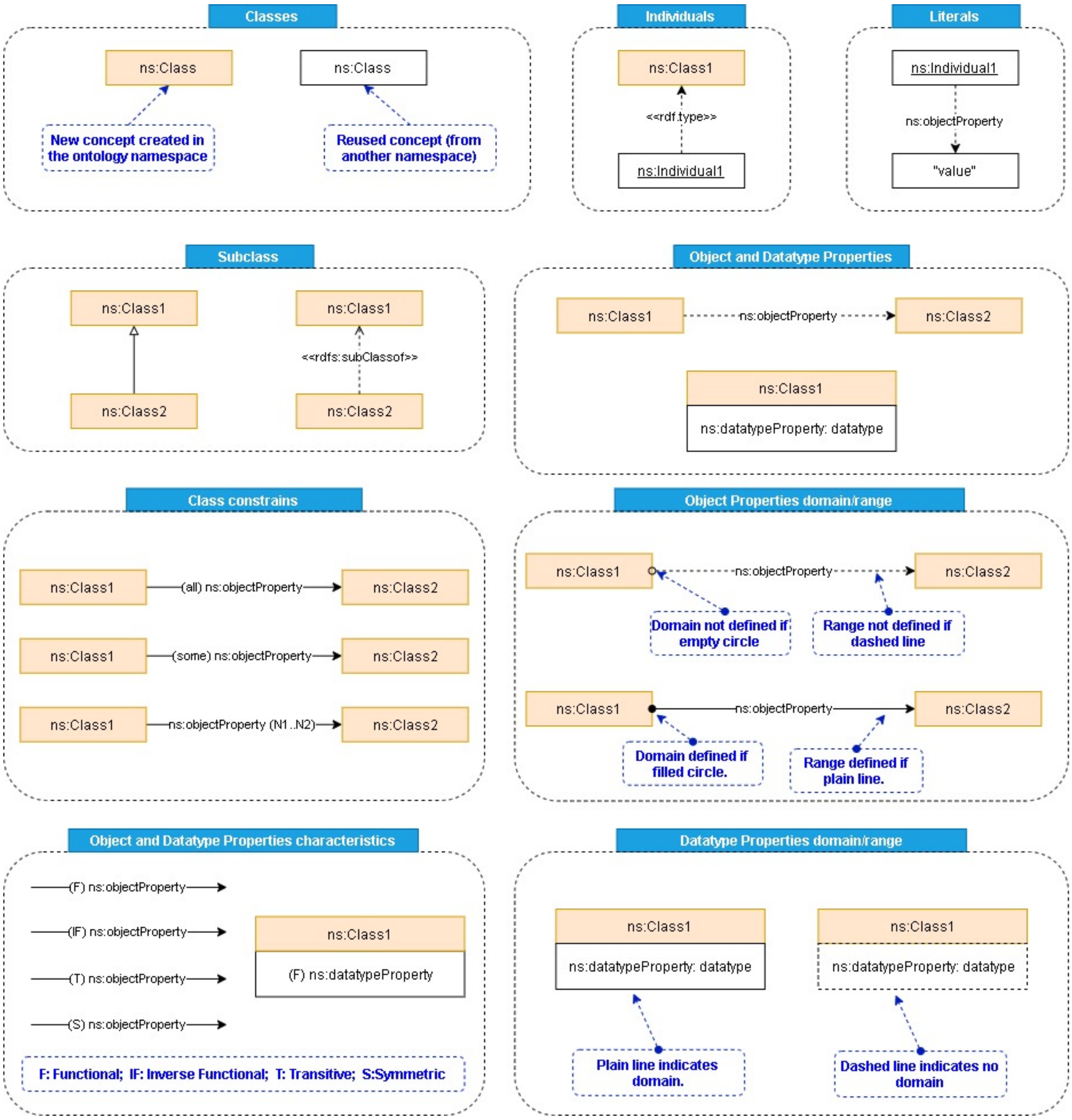


Figure 3. Visual Notation Summary.

2.2 THE BIMERR ONTOLOGY PORTAL

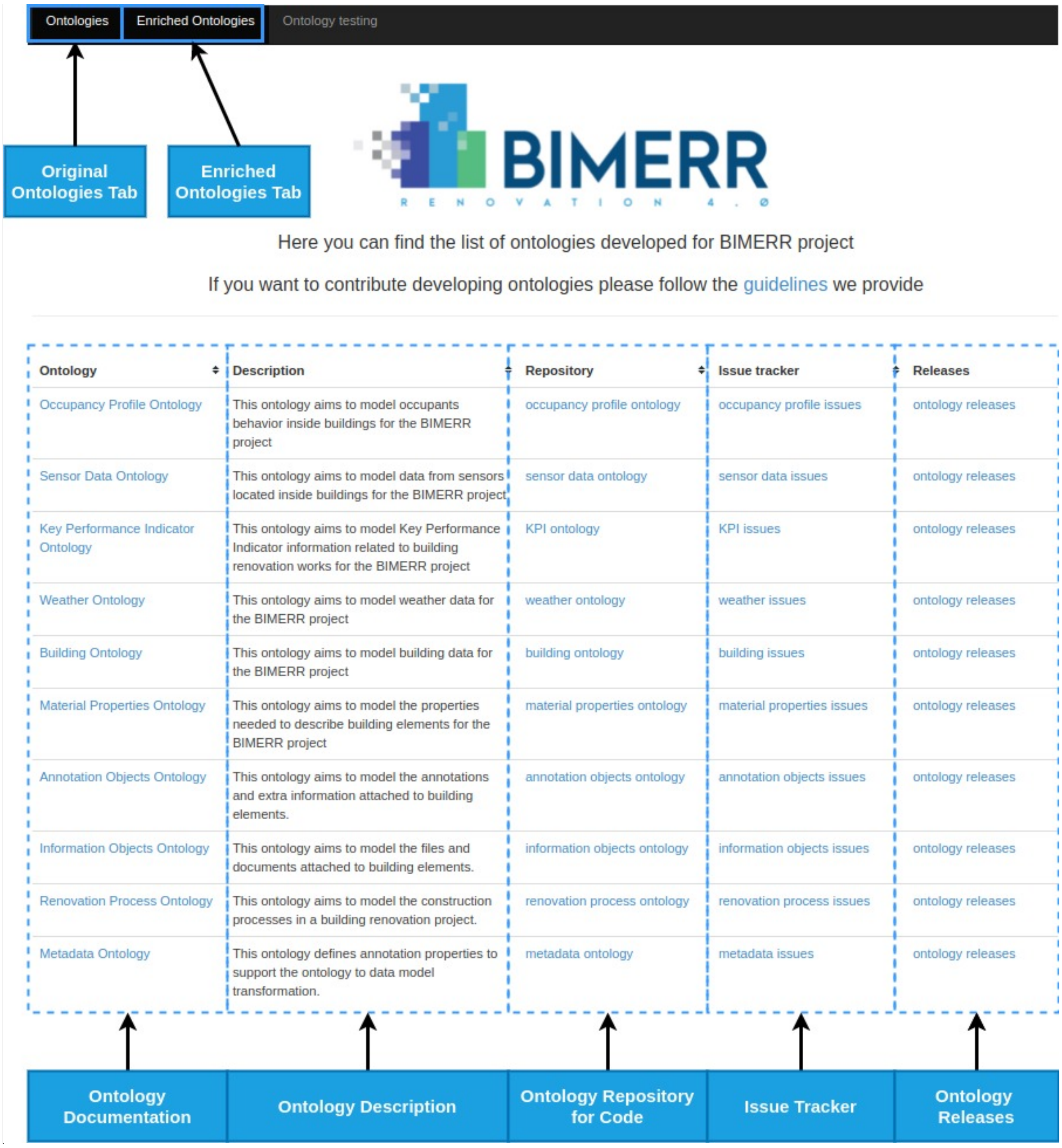
The BIMERR ontology portal gathers all the resources generated during the ontology development process. Figure 4 shows a snapshot of the current state of the portal, which is available at https://bimerr.iot.linkeddata.es/. The portal has two sections, each one providing different resources:

- Ontologies: It is the main section of the portal, which shows the main information about the ontologies. This section follows a tabular approach which includes:
    - Links to the ontologies' documentation published on the web. This documentation is generated automatically from the ontology implementation using the Widoco tool [Garijo et. al., 2020]. [10] The documentation also includes links that point to the code of the ontology.
    - An ontology description.
    - Links to the GitHub repositories containing the code and other resources for each model.
    - Links to the GitHub issue trackers.
    - Links to the GitHub releases.
- Enriched ontologies: It contains the ontologies with enriched metadata, and the data models obtained after the transformation process as we can see in Figure 5.

Regarding the guidelines section that can be accessed through the main section, it provides the users with a brief overview of the proposed process for developing ontologies and some guidelines which should be followed by the domain experts and ontology developers who wants to contribute. This section includes information about:

- How the repository should be structured;
- The tools recommended to be used during the ontology development process;
- Ontology versioning;
- Issues management.

[10] https://github.com/dgarijo/Widoco



Ontology	Description	Repository	Issue tracker	Releases
Occupancy Profile Ontology	This ontology aims to model occupants behavior inside buildings for the BIMERR project	occupancy profile ontology	occupancy profile issues	ontology releases
Sensor Data Ontology	This ontology aims to model data from sensors located inside buildings for the BIMERR project	sensor data ontology	sensor data issues	ontology releases
Key Performance Indicator Ontology	This ontology aims to model Key Performance Indicator information related to building renovation works for the BIMERR project	KPI ontology	KPI issues	ontology releases
Weather Ontology	This ontology aims to model weather data for the BIMERR project	weather ontology	weather issues	ontology releases
Building Ontology	This ontology aims to model building data for the BIMERR project	building ontology	building issues	ontology releases
Material Properties Ontology	This ontology aims to model the properties needed to describe building elements for the BIMERR project	material properties ontology	material properties issues	ontology releases
Annotation Objects Ontology	This ontology aims to model the annotations and extra information attached to building elements.	annotation objects ontology	annotation objects issues	ontology releases
Information Objects Ontology	This ontology aims to model the files and documents attached to building elements.	information objects ontology	information objects issues	ontology releases
Renovation Process Ontology	This ontology aims to model the construction processes in a building renovation project.	renovation process ontology	renovation process issues	ontology releases
Metadata Ontology	This ontology defines annotation properties to support the ontology to data model transformation.	metadata ontology	metadata issues	ontology releases

Figure 4. BIMERR Ontology Portal

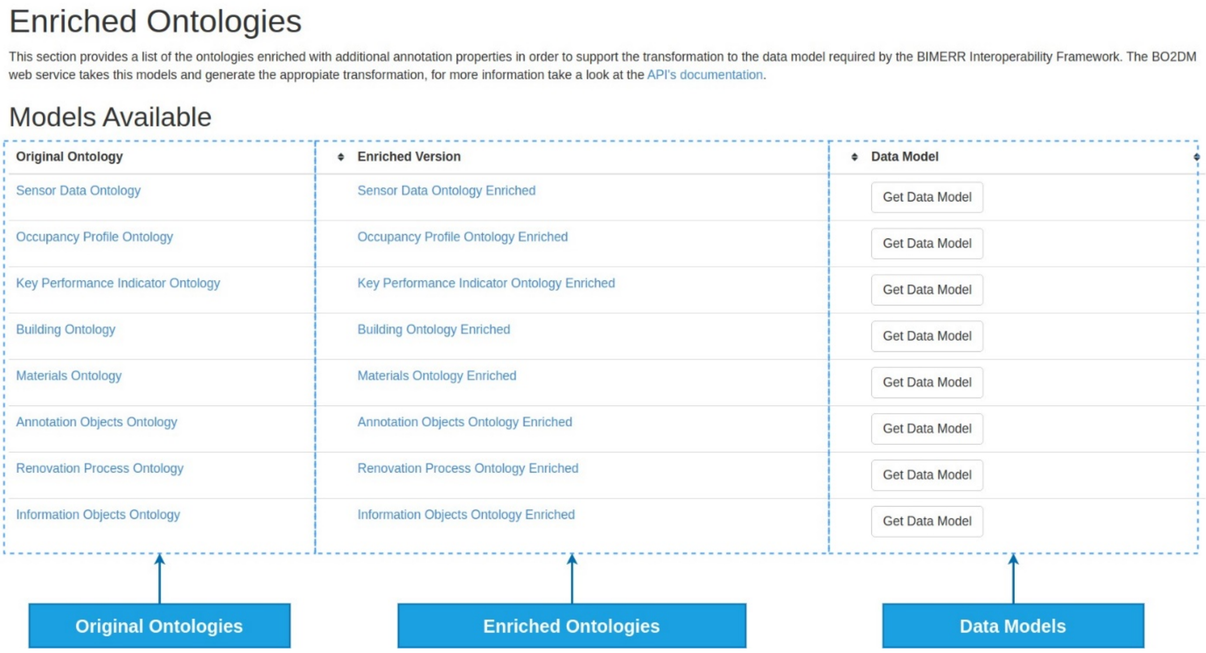


Figure 5. BIMERR Ontology Portal – Enriched Ontologies Section

2.3 ONTOLOGY RESOURCES

The resources generated during the process of ontology conceptualization, implementation and publication are centralized in GitHub repositories created for each individual domain, as it can be seen in Figure 6. The information contained in the repositories is structured as follows:

- **The diagrams folder:** Contains the conceptualization diagrams of the developed ontology in an XML format.
- **The documentation folder:** Contains the HTML documentation of the model.
- **The ontology folder:** Contains the implementation of the ontology in Turtle format.
- **The requirements folder:** Contains the requirements of the ontology in tables and the ORSD (Ontology Requirements Specification Document) for each module.

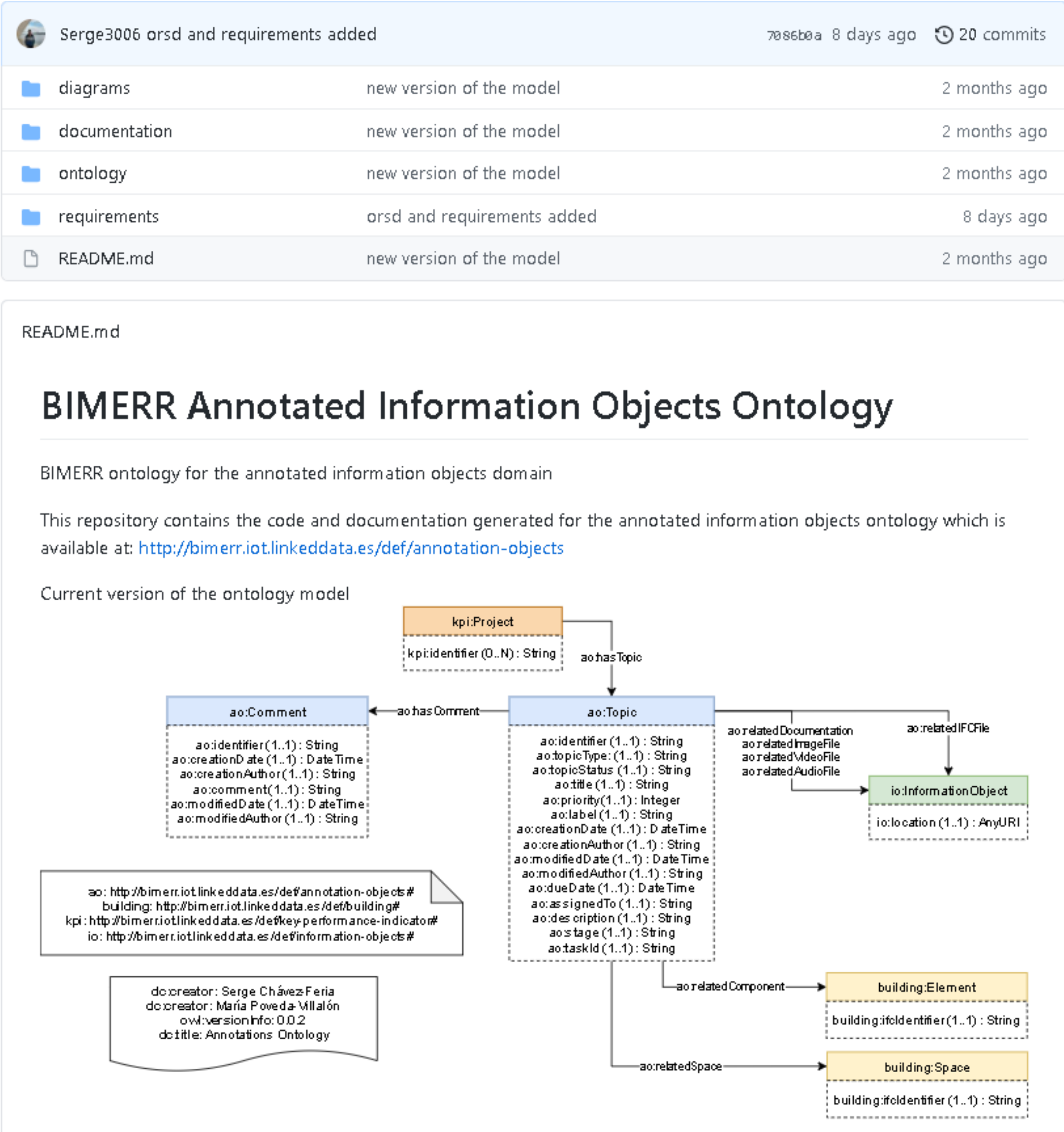


Figure 6. GitHub Repository of the Annotation Objects Ontology

Additionally, the GitHub repositories support the maintenance of the ontology, by managing the list of issues identified by the domain experts and ontology developers.

All the changes and improvements to the ontology need to be agreed among all members of the ontology development team. The GitHub issue tracker is used to discuss improvements and issues about the domains. If domain experts or ontology developers want to add new concepts to the ontology, they have to create a new issue in the GitHub issue tracker, which will be used to start a discussion about the approval of the proposal.

GitHub issues are marked as "open" and "closed". The issues can also have an assignee, which is the person that is responsible for moving the issue forward. Figure 7 shows the issues section associated to the Building ontology.

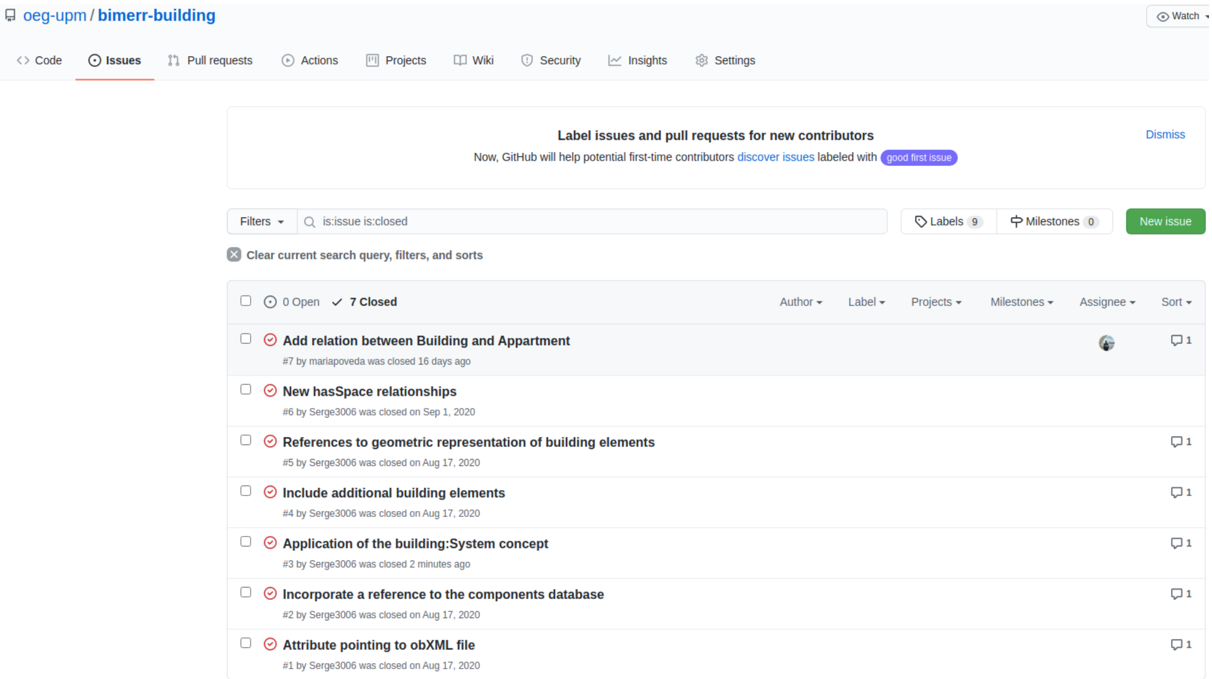


Figure 7. GitHub Issues Tracker for the Building Ontology

On the other hand, the publication of the HTML documentation of the ontologies is handled by UPM servers. The documentation of the models includes general information about the ontology (version, publisher, creators, etc.), an introduction about the purpose of the ontology, a detailed description of the model, and the list of concepts and properties along with their corresponding definitions. Figure 8 shows an example of the documentation generated for the building ontology.

The fact that the documentation is stored on GitHub allows the automation of the deployment. Every time a new version of the ontology is tagged as a release in GitHub, it communicates with the Ontology Manager Framework, which automatically pulls the documentation stored in the repository, displaying the updated information in the site.

Ontology Specification Draft

language en

Building ontology

Ontology URI:
http://bimerr.iot.linkeddata.es/def/building#

Revision:
0.1.5

Authors:
Serge Chávez-Feria (Ontology Engineering Group, Universidad Politécnica de Madrid)
María Poveda Villalón (Ontology Engineering Group, Universidad Politécnica de Madrid)

Publisher:
http://www.oeg-upm.net/

Download serialization:
Format JSON LD | Format RDF/XML | Format N Triples | Format TTL

License:
License http://purl.org/NET/rdflicense/cc by4.0

Cite as:
Serge Chávez-Feria, María Poveda-Villalón. Building Ontology: v0.1.1

Provenance of this page

Abstract

The Building ontology has been developed to perform as the core module of the BIMERR Ontology Network, containing information related to building topology, and components. The model is constructed as an extention of the BOT ontology that provides the vocabulary to describe the topology of a building as well as the relationships between their main components such as zones, spaces, and building elements. The taxonomy of building components is based on the proposed by the IFC 4.1 standard.

Figure 8. Excerpt of the Building Ontology HTML documentation

2.4 DATA MODELS

The JSON data models generated from the ontologies can be obtained from the BIMERR ontologies portal (see Figure 9). When a new version of the ontology is generated, the ontology engineers need to manually update the data models stored in this section.

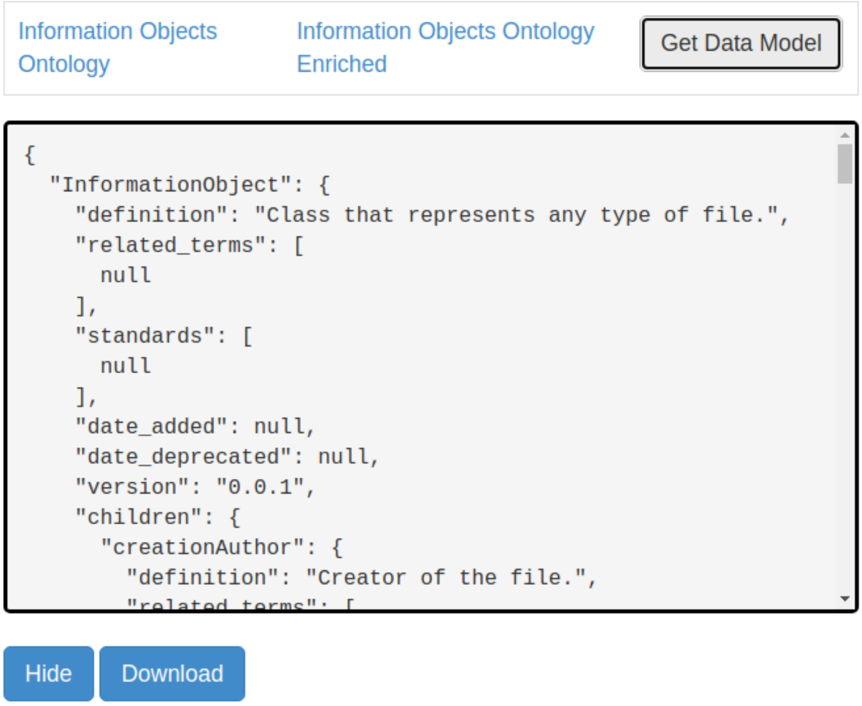


Figure 9. Data Models Stored in the Portal

3 BIMERR ONTOLOGY NETWORK DESCRIPTION

In this section, the ontologies that composes the BIMERR network are thoroughly presented. The section provides a description of the usability of the model, an explanation of the conceptualization, example(s) of how to instantiate it, and a list of all the resources generated during the its implementation. Additionally, the links between the individual models are explained and some decisions made during its development are described.

3.1 BUILDING MODULE

The purpose of the Building Ontology is to represent the main spatial topological elements of a building. Furthermore, the model aims to represent the main components and systems that are directly involved in energy related aspects of buildings. This model acts as the core module within the BIMERR Ontology Network, where several other domains related to the building industry are connected to it. The model is constructed as an extension of the BOT ontology [Rasmussen et. al., 2020] that provides the vocabulary to describe the topology of a building as well as the relationships between their main components such as storey, zones, spaces, and building elements. Additionally, it reuses the building elements taxonomy from the SAREF4Building Ontology [Poveda-Villalón et. al., 2018], a SAREF extension reusing the device hierarchy from IFC 4.1. This hierarchy was used to represent the components and systems that directly impact the energy consumption of buildings and are susceptible to change in a renovation project.

Figure 10 presents the main elements of the current state of the building model, their attributes as well as the relationships between them. We can see that the topological relationships flow from the most generic entity, that is `building:Building` to the most specific concept, which is `building:Element`, passing through the concepts of `building:Apartment`, `building:Storey`, and `building:Space`. However, we have also added some shortcut relationships between the concepts according to the requirements of the project.

Figure 11 depicts the classification of building elements to be used under the scope of renovation projects. As mentioned before, most of the concepts in this hierarchy are reused from the SAREF4Building ontology, but we include only those that are relevant to the project according to the use cases.

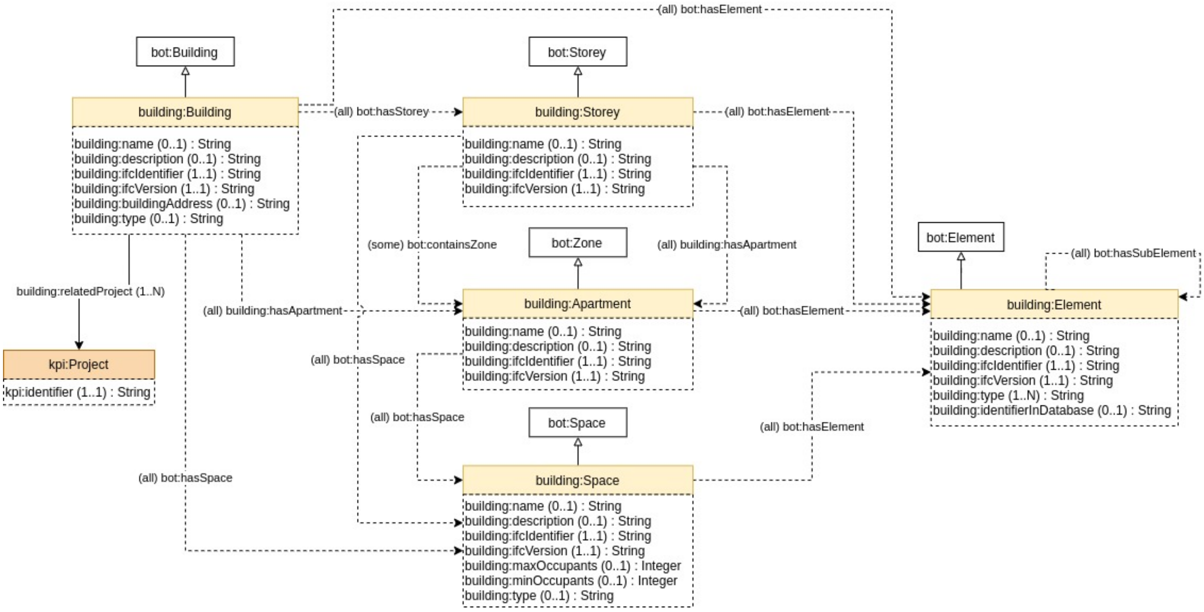


Figure 10. Building Ontology Conceptualization - Topology

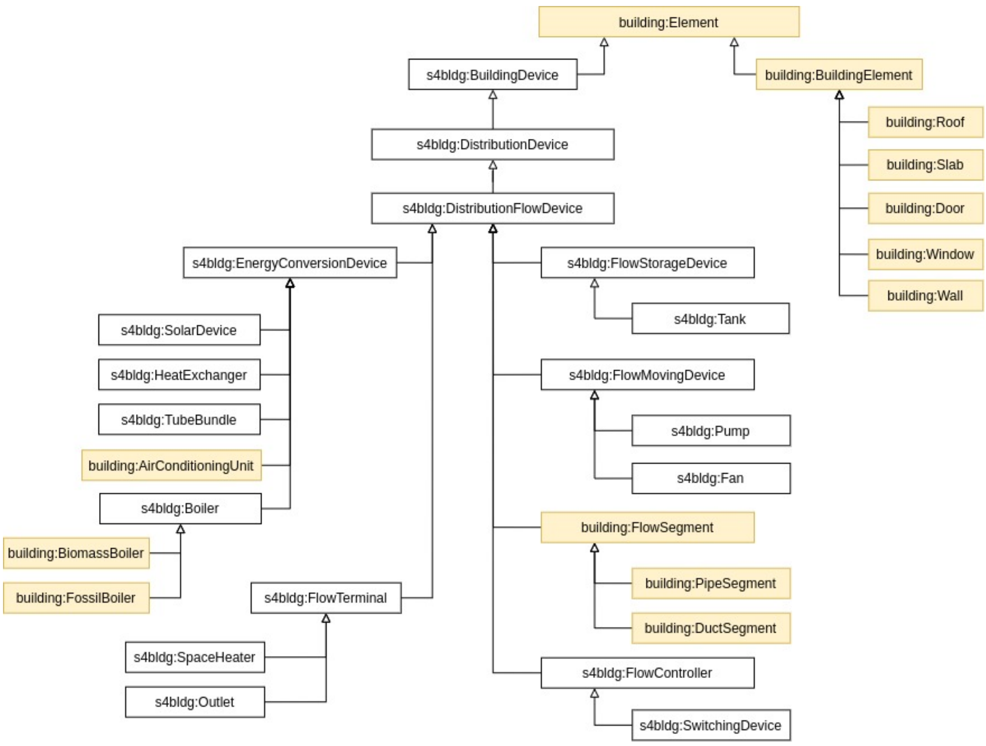


Figure 11. Building Ontology Conceptualization – Taxonomy of Building Elements

Apart from the main concepts shown before the model allows the instantiation of information regarding the building element properties and their corresponding values. Figure 12 shows this part of the model that reuses some terms and design patterns from the SAREF ontology. This section of the model is justified by the fact that some tools such as BICA will be used to report on properties of building elements such as HVAC components.

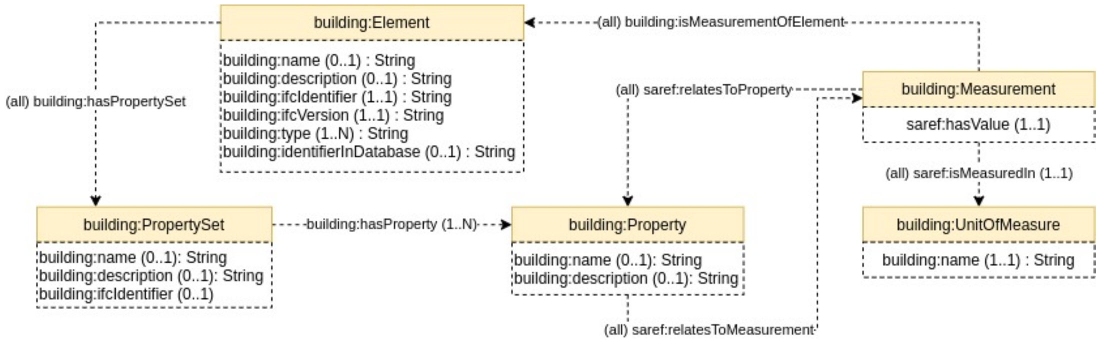


Figure 12. Building Ontology Conceptualization - Properties

3.1.1 Model Description

The most important concepts related to the Building domain are the following:

- **Building:** Represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, storey, and space).
- **Storey:** The building storey has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound.
- **Space:** A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.
- **Zone:** A part of the physical world or a virtual world that is inherently both located in this world and having a 3D spatial extent.
- **Apartment:** A self-contained housing unit which is composed by several spaces.
- **Element:** An element is a generalization of all components that make up an AEC product.

The topological elements shown in Figure 10 are: building:Building, building:Storey, building:Apartment, and building:Space. Each of these elements should have an IFC identifier that uniquely identify them in the IFC model and corresponds to the IfcGlobalId attribute in that schema. The model builds around the root concept building:Building, that incorporates attributes to indicate its physical location by means of geospatial coordinates (wgs84_pos:long, wgs84_pos:lat, and wgs84_pos:alt) provided by the W3C geo vocabulary. The relationships between the topological elements are taken directly from BOT. A building may have several stories, which in turn may contain one or more residential apartments. These relationships are materialized by the object properties bot:hasStorey and building:hasApartment respectively. An apartment has different spaces such as a living room, a kitchen or bedroom, and these spaces comprise of building components that delimit the zone itself (building:Door, building:Slab, building:Window, and building:Wall) and by energy related equipment (building:SpaceHeater or building:Fan). Some applications within the BIMERR solution require the original IFC representation of the topological elements. The model allows the connection to those files by means of the building:hasIFCFile relation, which points to an internal location within BIF.

Another aspect of the model is the classification of elements that are typically considered in building renovation projects focused on energy efficiency. Those elements are represented in the model by the building:Element concept, which is also included in the IFC schema. Usually, these elements have an impact over the energy performance of buildings. The elements classification used in this ontology (concepts under the super-class building:Element in Figure 11) follows the structure proposed by the IFC 4 standard.

Additionally, the building elements can be further described using properties (see Figure 12). The conceptualization of properties follows the design pattern of IFC, where properties of the same type or typical to specific elements are grouped into property sets. Also, for the representation of the property values the design pattern from the SAREF Ontology is reused, where each property is related to a measurement, which has an associated a unit of measure.

3.1.2 *Data Example*

Figure 13 presents an example of how to utilize the ontology for a fictional small building. The building instance identified in the model as `data:Building01` has an IFC identifier "2cZtnlll9DugwKjW2uDsyk" extracted from the IFC file, which is linked to this entity using the `building:hasIFCFile` relationship. The building is located at a latitude of 41.730330°, and a longitude of -2.441711°.

The building has two stories. The first storey (`data:Storey01`) has two apartments (zones), where each one has several spaces. To avoid an overwhelming graph, the figure just represents some elements. The first apartment (`data:Apartment01`) has two spaces: the kitchen (`data:Space01`) and the bedroom (`data:Space02`). The kitchen includes some building elements such as doors, slabs and walls. The second storey (`data:Storey02`) contains one apartment (`data:Apartment03`), where only the living room (`data:Space03`) is represented. This space contains some energy related devices such as a space heater and an outlet.

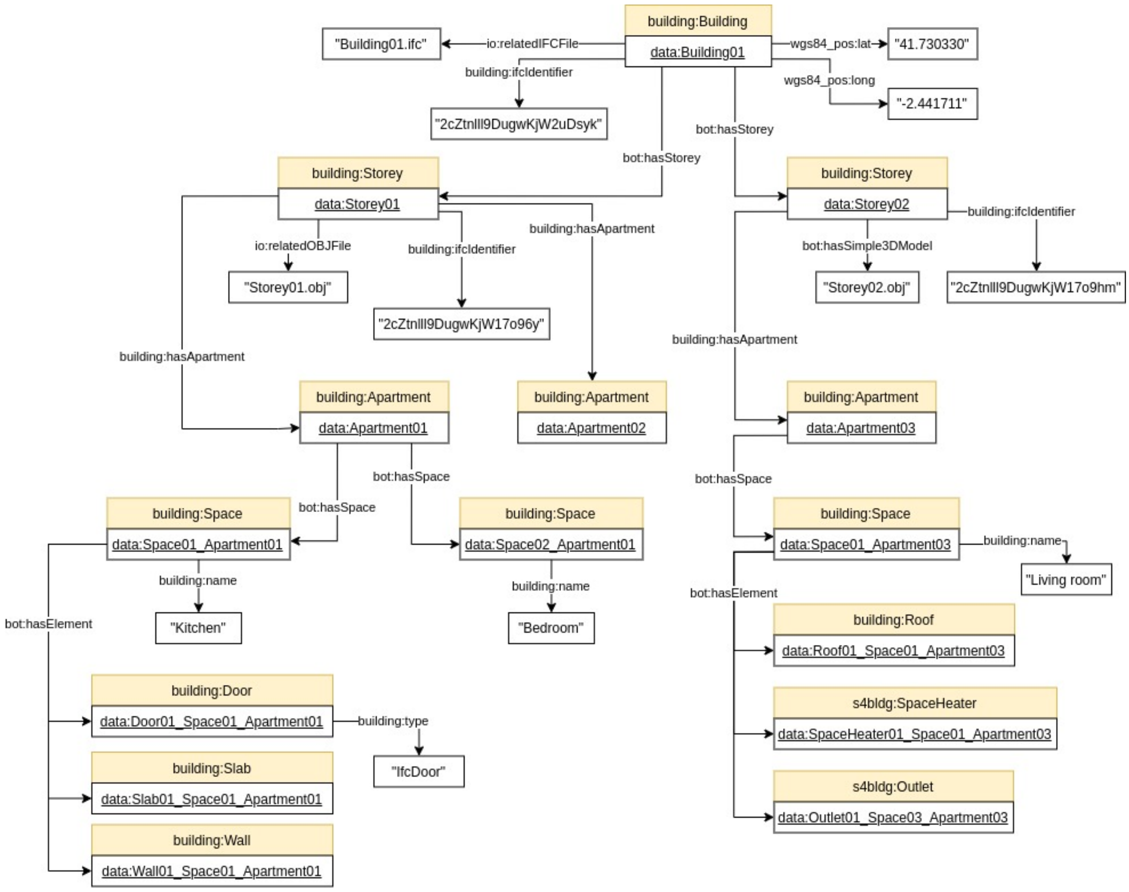


Figure 13. Data Example of the Building Ontology

3.1.3 Model Evaluation

The building ontology has been evaluated in two different ways. The first method consists in the detection of common pitfalls in the ontology implementation. Such pitfalls are identified using the OOPS! Web application [Poveda-Villalón et. al., 2014].[11] The results of the evaluation of this model are illustrated in Figure 14.

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

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 P10: Missing disjointness.	**ontology* | Important**
Results for P11: Missing domain or range in properties.	**20 cases | Important**
Results for P13: Inverse relationships not explicitly declared.	**15 cases | Minor**
Results for P24: Using recursive definitions.	**1 case | Important**
Results for P40: Namespace hijacking.	**1 case | Critical**
Results for P41: No license declared.	**ontology* | Important**
SUGGESTION: symmetric or transitive object properties.	**2 cases**

Figure 14. OOPS results for the Building Ontology – First Evaluation

It can be seen that there are several types of errors classified according to their importance level. The critical errors are crucial to correct; otherwise it would affect several aspects of the ontology performance. The important errors are not critical but are important to correct. The minor ones are not actual problems but suggestions to make the ontology nicer.

In this particular case, the building ontology has one critical error pointing that we are using the incorrect namespace for the term `dc:license`. By changing the URI of the term to the correct one http://purl.org/dc/terms/license, the error was solved. The important cases are mostly related to missing domain and range specification in the object properties and datatype properties of the ontology. However, not all the properties need a domain and range specification because some of them are used by several concepts; thereby, only some properties were modified. Finally, the minor errors section indicated the lack of inverse relationships which is something that is not addressed because there is no need or use case which supports the inclusion of those relationships in the model.

The evaluation results after executing OOPS! for a second time are shown in Figure 15. This time we can see that the critical errors has been solved and that the number of important errors has been reduced to just nine cases.

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 P11: Missing domain or range in properties.	**9 cases | Important**
Results for P13: Inverse relationships not explicitly declared.	**15 cases | Minor**
Results for P24: Using recursive definitions.	**1 case | Important**
SUGGESTION: symmetric or transitive object properties.	**2 cases**

Figure 15. OOPS results for the Building Ontology – Second evaluation

The same process followed using OOPS! in this section is going to be repeated for the rest of the models, ensuring that no critical errors exist, and correcting important errors if appropriate. This process will not be described in the subsequent models in order to avoid repetitive information.

The second method evaluates the suitability of the model for a specific application [Sabou et. al., 2012], which in this case is the BIMERR Interoperability Framework. It is needed to evaluate if the model fulfills the requirements imposed from the BIF side in order to match the platform functionalities or its special features. In the specific case of the building data model additional properties were included. For instance, in order to avoid ambiguities with respect to the version of the IFC standard being used, the `building:ifcVersion` attribute was attached under the concepts that can be described by means of an IFC file. Also, the `building:type` attribute was added in the `building:Element` concept in order to facilitate the querying of building components. Finally, the section of the ontology related to the description of properties and measurements has been replicated from the material properties ontology. This is a temporary modeling decision until the linking capabilities between models of BIF has been tested.

3.1.4 Links to other models

The building ontology is linked to two other modules: The KPI module and the Information Objects module. In order to discover which renovation projects have been performed over a building entity, the model includes the relation `building:relatedProject` between the concepts `building:Building` and `kpi:Project`.

Furthermore, we can link any topological element of the building to additional documents using the `io:relatedDocumentation` object property. We can be even more specific and ask for IFC files (`io:relatedIFCFile`) or OBJ files (`io:relatedOBJFile`).

3.1.5 Model Particularities

The structure of the data models finally expected by the BIMERR Interoperability Framework imposes some restrictions and requirements over the ontologies developed in the project.

In the particular case of the building ontology additional attributes and relations were added in order to facilitate the population and querying of building information. For instance, the `building:type` attribute was added to query building elements by its type. Even though, this operation is naturally supported by ontologies implemented in the OWL language, we need to include those artificial fields to allow those queries in BIF.

3.1.6 List of Resources

The following list of resources are provided for the building ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/building/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-building
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/building/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/building_enriched.ttl

3.2 ANNOTATION OBJECTS MODULE

The Annotations Objects Ontology aims to represent the annotations produced during the development of a building renovation project. These annotations inform about issues found during the renovation activities or missing information not gathered during the surveying process that could be relevant for the project, such as missing building components. The model is based in the Building Collaboration Format (BCF), which is the standard for the representation of issues between different BIM modeling tools. Figure 16 presents an overview of the concepts, relations and attributes included in the model.

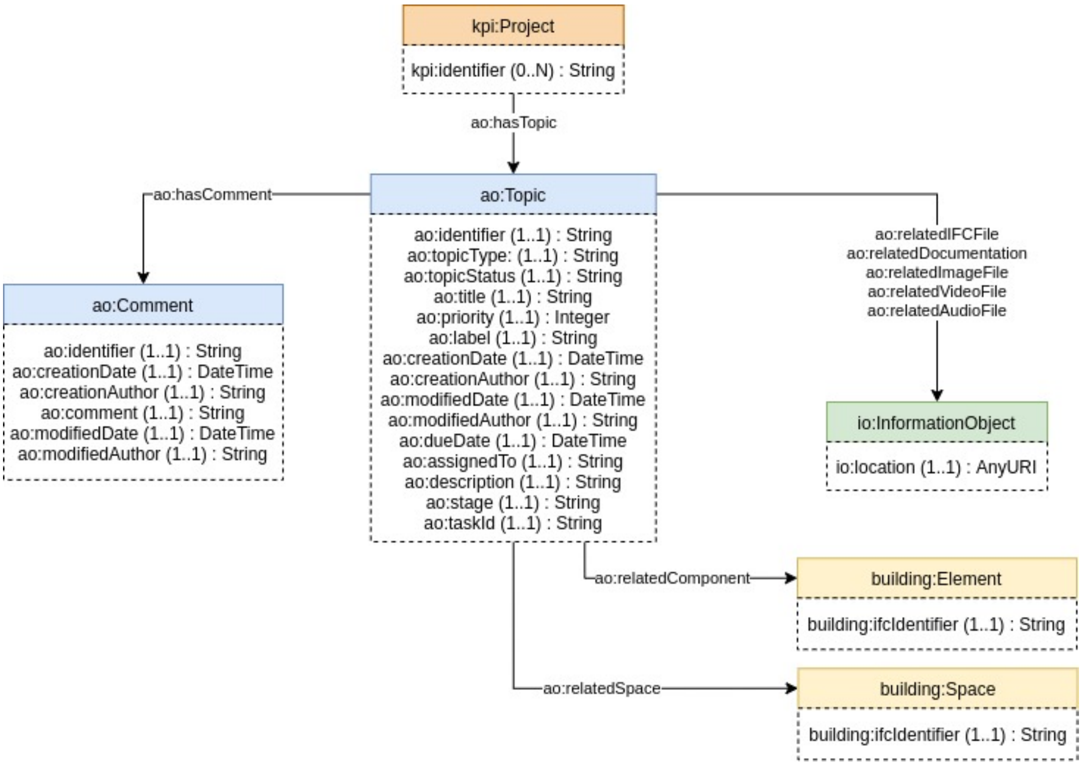


Figure 16. Annotations Ontology Conceptualization

3.2.1 Model description

The main concepts in the model are `ao:Topic` and `ao:Comment`, which are inherited from the BCF standard. The `ao:Topic` represents reference information about the topic of the issue or annotation. We can describe those topics using attributes such as `ao:identifier` or `ao:title`. We can categorize those topics according to different aspect of the project:

- **Stage (**`ao:stage`**):** Preliminary Planning End, Construction Start, Construction End, etc.
- **Label (**`ao:label`**):** Architecture, Structure, Mechanical, Electrical, etc.

- **Status (**`ao:topicStatus`**):** Open, In Progress, Closed, or ReOpened.
- **Type (**`ao:topicType`**):** Comment, Issue, Request, or Solution.

Additionally, the model gives the possibility to examine the full history of the modifications of this topic (`ao:creationDate` and `ao:modifiedDate`) and the authors of those changes (`ao:creationAuthor` and `ao:modifiedAuthor`), as well as assign an specific worker to solve the issue (`ao:assignedTo`).

Finally, it allows to add comments (`ao:Comments`) related to the topic at hand. The purpose of those comments is to record the discussion between different stakeholders and organization related to the topic. Similar to the case of topics we can keep a history of the modifications of those comments using the same fields.

3.2.2 Data Example

Figure 17 illustrates a simple example of how to populate the ontology. In this case, we have the Topic `data:Topic01` create by the Worker 01 related to electrical issues. This topic was created during the surveying stage of the project before the renovation activities start. It was categorized as an issue and is related to the electrical discipline. The topic has two comments linked to it, where `data:Comment01` was create by the same author of the topic and describes that some electrical cables are hanging from the roof of one of the stories. The second comment `data:Comment02` was created after two days just to confirm that the issue was solved.

The topic is further documented with images (`data:ImageFile01`) from the site and it is related to specific building components of the BIM model, in this case the `data:Roof02`.

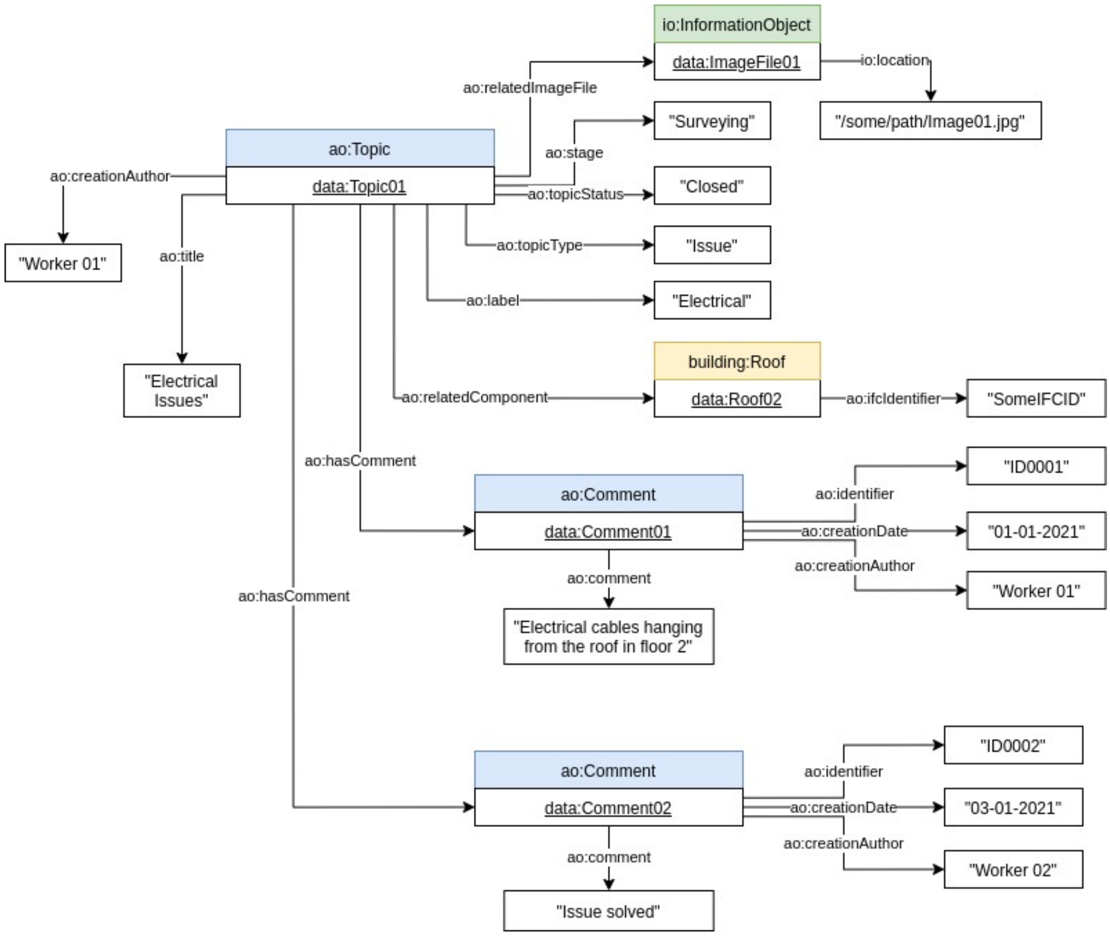


Figure 17. Data Example of the Annotation Objects Ontology

3.2.3 Model Evaluation

The model has been evaluated using OOPS!, the detailed results are available at the ontology repository.

3.2.4 Links to other models

During the process of generating annotations, one worker may want to attach complementary information to the textual information that would better describe the problem. The model covers this requirement by linking the topics with the Information Objects ontology. The `ao:InformationObjects` concept represents any file that is stored in BIF. It is possible to make the connection between the topics and the files using any of the object properties defined in the model depending on the type of file to be referenced. For instance, one could use the `ao:relatedImageFile` to indicate the location of an image or the more general one `ao:relatedDocumentation` for file types not considered in the model.

The ontology is also connected to the building module to indicate to which topological elements (`building:Space` or `building:Element`) the topic refers to. Also a connection to the `ao:Project` concept from the KPI ontology is provided to allow the retrieval of all the issues related to the project.

3.2.5 Model Particularities

As mentioned in the description section, we can categorize a topic using several classifications. For instance, we can classify it by the stage of the project it was taken, or by the type of issue it tries to address. We include that in the model by means of attributes attached at the Topic level. Each of those attributes can contain a set of default values, which can be naturally implemented in the ontology using RDF collections. However, due to the current limitation of BIF to not accept lists as a datatype, the values for those fields are left empty, just indicating that the values should be strings.

3.2.6 List of Resources

The following list of resources are provided for the annotation objects ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/annotation-objects/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-annotation-objects
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/annotation-objects/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/ao_enriched.ttl

3.3 RENOVATION PROCESSES MODULE

The purpose of the Renovation Processes Ontology is to represent the different processes and tasks to be executed during the implementation of a building renovation project, as well as the work orders received by the workers onsite.

During the planning stage of a project a detailed workflow is developed with the sequence of processes needed in order to execute it successfully. Each process is divided into smaller tasks that are assigned to specific workers. Those workers are notified about the activities they have to perform and report back the progress done in those activities.

3.3.1 Model description

The conceptualization of the ontology is shown in Figure 18. The renp:Process represents the processes that are specified on the workflow plan. Those processes can be further divided into smaller tasks (renp:Task). Both concepts can be assigned to one or more workers (renp:Worker) using the relation renp:executedBy in order to indicate who is the responsible for its execution. Apart from those core elements, we can connect the renp:Process and renp:Task with key performance indicators in order to communicated them to the workers on site so they are aware of the objectives to achieve.

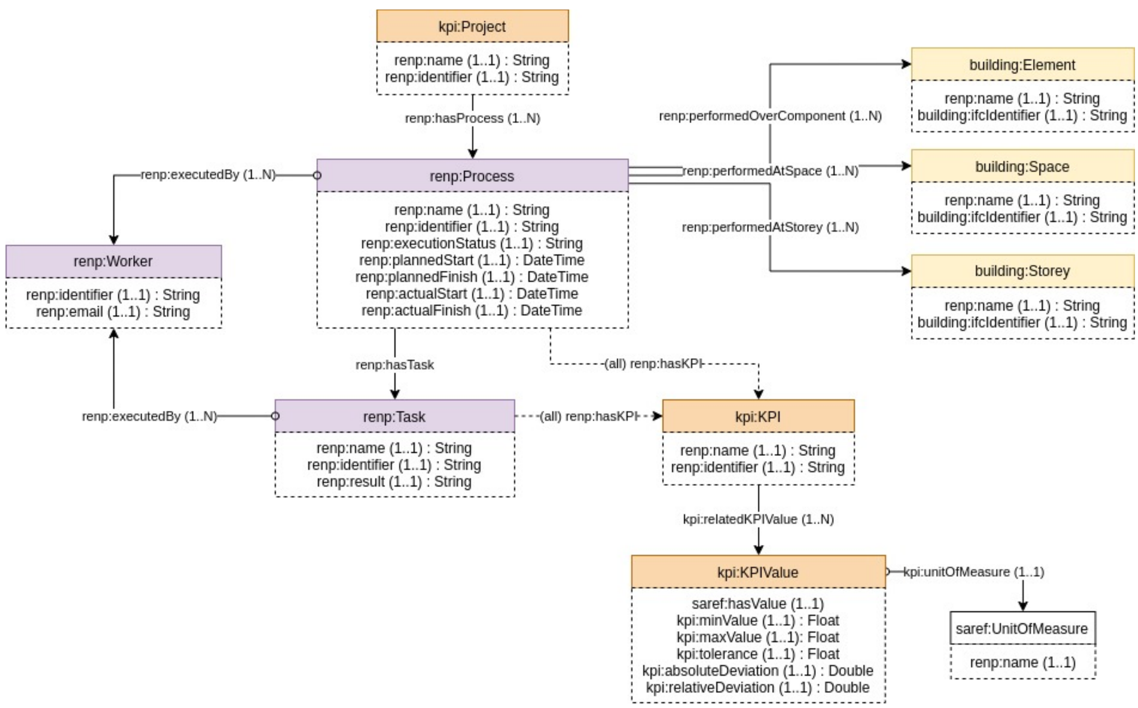


Figure 18. Renovation Processes Ontology Conceptualization

The processes are further detailed using a series of attributes. It is possible to indicate which are the planned finishing and starting times (`renp:plannedStart`, `renp:plannedFinish`), and compare them to the actual figures on those metrics (`renp:actualStart` and `renp:actualFinish`). The model also allows the instantiation of the actual process on the process at hand by means of the datatype property `renp:executionStatus`.

In order to facilitate the querying of processes and tasks from the worker's side, the processes are connected to the topological elements of the building. This connection allows to know in which spaces (`renp:performedAtSpace`) or over which elements (`renp:performedOverComponent`) the renovation activities need to be performed.

3.3.2 Data Example

Figure 19 shows an example of how we can populate renovation process ontology. The project `data:Project01` has just one process (`data:Process001`) called "Facade Improvement Process". This process is composed by two tasks. The first task is the installation of the material lift and crane (`data:Task001`) followed by the installation of safety measures (`data:Task002`). Both task are connected sequentially by the `renp:hasNextTask` relation. The initial plan considered that `data:Task001` should start at 2020-01-01 and should finish at 2020-01-20; however, the task took more time and the actual finished time was 10 days after the initial planned date. The completion of `data:Task001` gives permission for the start of the second task (`data:Task002`) that was planned for 2020-02-01 and currently it is only completed at 50%.

The installation of the material lift and crane was performed by the `data:Worker01` with email email@company.com and personal identification "SomeInternalID".

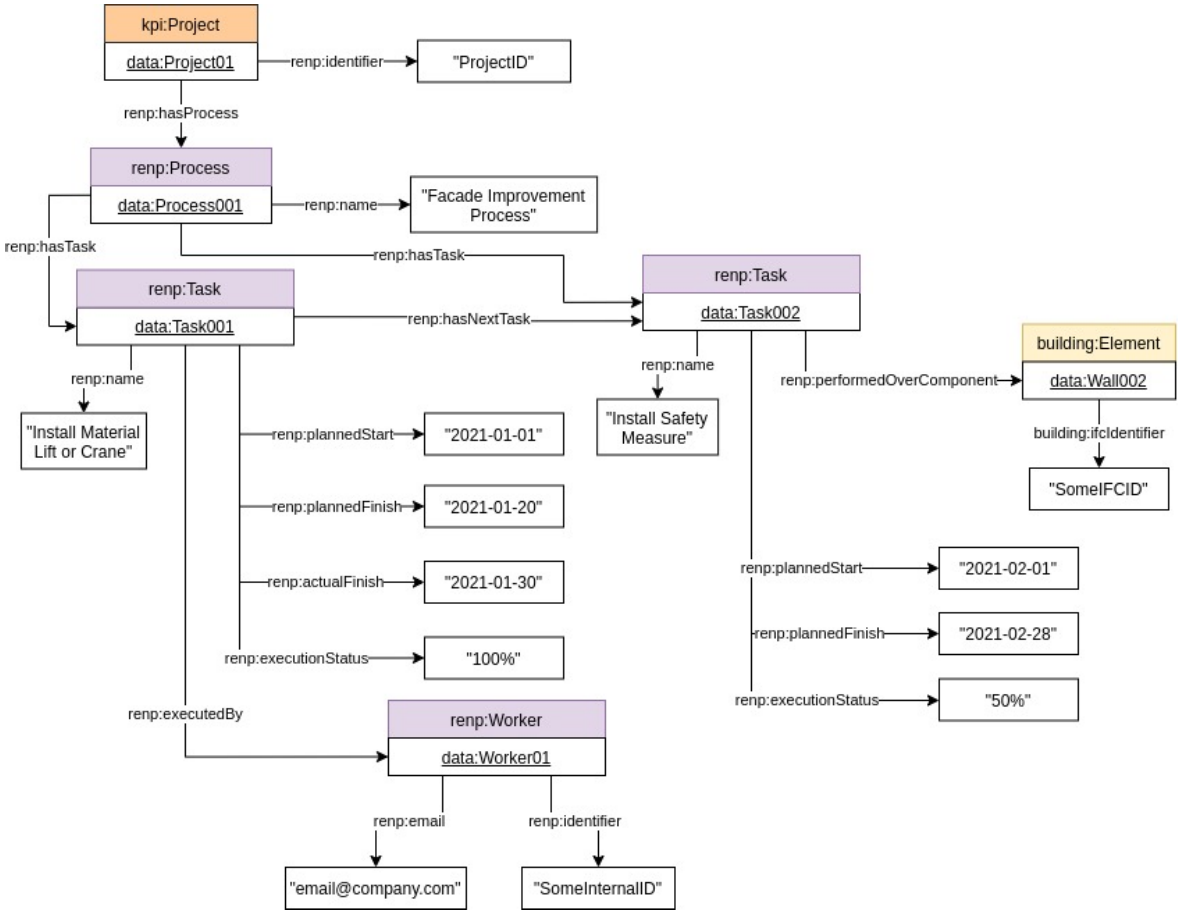


Figure 19. Data Example of the Renovation Process Ontology

3.3.3 *Model Evaluation*

The model has been evaluated using OOPS!, the detailed results are available at the ontology repository.

3.3.4 *Links to other models*

The renovation process model is connected to two other modules in the BIMERR Ontology Network. The first connection is with the Key Performance Indicators Ontology using the `kpi:Project` and `kpi:KPI` concepts. By making this connection we can facilitate the retrieval of all the processes related to a project, and the specification of planned and calculated KPI's by activities. The second connection is with the Building ontology, specifically with the following concepts: `building:Space`, `building:Element`, and `building:Storey`. As discussed earlier these links implemented at the process level enable the identification of all the activities that need to be executed by location or by building element.

3.3.5 Model Particularities

The ontology includes concepts from the KPI ontology in order to support the use cases that deal with the notification of those metrics to workers on site. Some concepts from the KPI module were replicated in this ontology such as the `kpi:KPIValue` class to instantiate specific values for these indicators, as well as the `saref:UnitOfMeasure` concept to represent the units of these metrics.

3.3.6 List of Resources

The following list of resources are provided for the renovation process ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/renovation-process/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-renovation-process
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/renovation-process/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/renp_enriched.ttl

3.4 INFORMATION OBJECTS MODULE

The purpose of the Information Objects Ontology is to represent the files and documents generated during a building renovation project and are stored as information objects in the BIF (BIMERR Interoperability Framework). Figure 20 shows the concepts and relations involved in the model.

During the information exchange process between the renovation tools there are certain types of files that do not require a very detailed description or representation of their data models in the BIF. This feature is basically driven by the fact that some tools do not require specific information from those files, but they need the document as a whole in its original format. Apart from that, there is a need to keep a track of the history of the information generated during the project. Some of the files that can be represented by the ontology are BIM files (IFC or OBJ files), obXML files, BPMN plans, PDF specifications, CAD files, images, etc.

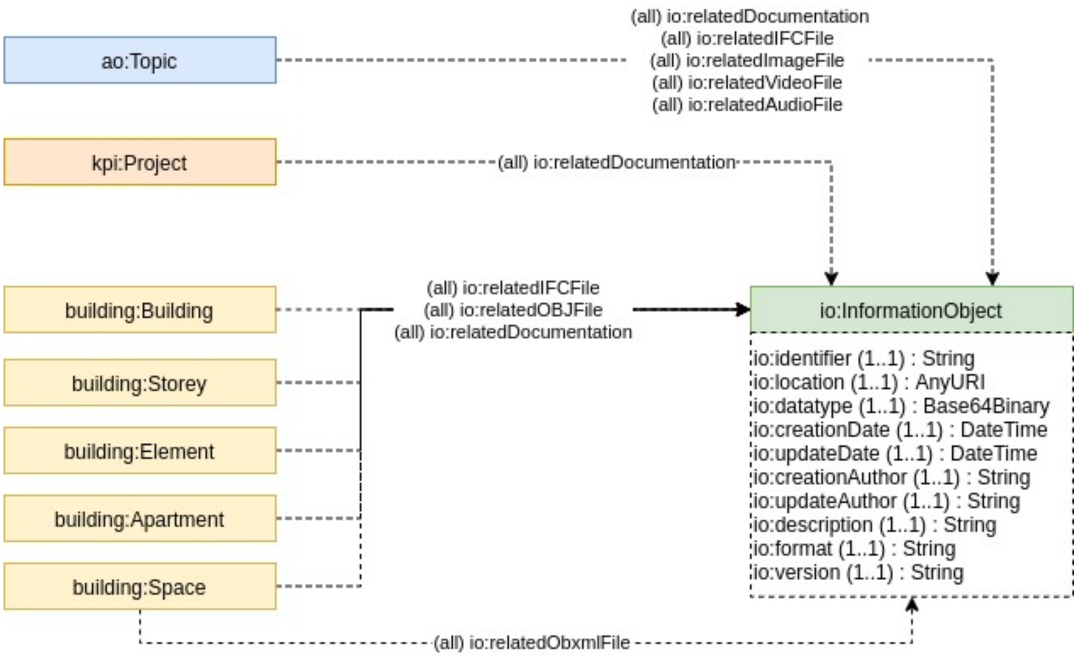


Figure 20. Information Objects Ontology Conceptualization

3.4.1 Model description

The main concept in this ontology is the `io:InformationObject` class, that represents all the files and documents that are stored inside the BIMERR Interoperability Framework. By using this entity to annotate our documents it is possible to add additional metadata to indicate attributes such as the location (`io:location`), the version of the document (`io:version`), the creator (`io:creator`) or the creation date (`io:creationDate`). The inclusion of these datatype properties allows the execution of more complex queries when searching for documents inside the BIMERR platform.

Furthermore, the ontology includes relations such as `io:relatedDocumentation` that allows that classes from other domains could also be linked to documents.

3.4.2 *Data Example*

Figure 21 shows an example on how to use this ontology. In this use case it is possible to have the project `data:Project01` and the building `data:Building01` that are connected to a BIM information object using the object property `data:relatedIFCFile`. The BIM object in this example is called `data:Building01_3DModel` and is located within BIF in the internal path "/some/path/Building.ifc". The responsible for the creation of the file in the platform was the "Modeler01" and it was created on 2018-03-27. The file also include a description with additional comments indicating that the BIM file was obtained as a result of the surveying process of the building. Finally, additional metadata is instantiated in order to facilitate the discovery and retrieval of this file such the format (.ifc) and the current version (0.0.1).

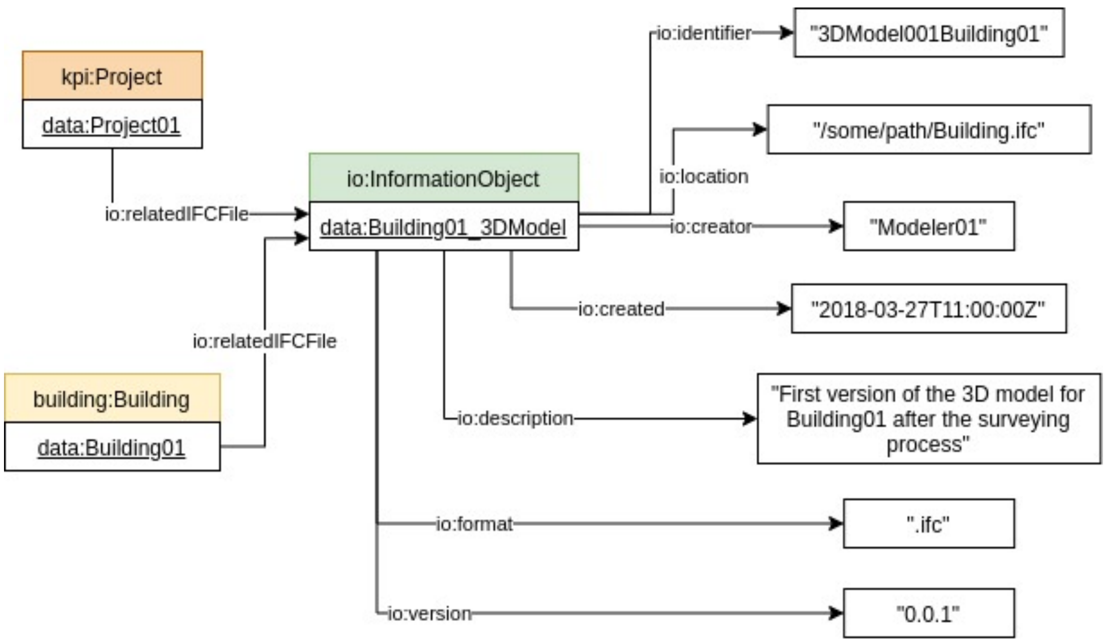


Figure 21. Data Example for the Information Objects Ontology

3.4.3 *Model Evaluation*

The model has been evaluated using OOPS!, the detailed results are available at the ontology repository. It is important to remark that this model has as main objective the identification of a file inside BIF, therefore it is not validated with the domain data input of any of the applications in the project.

3.4.4 Links to other models

As it has been seen in the previous modules, the information objects ontology is the one that has more links with the rest of the models due to the universal requirement to attach documents to the different concepts that compose the BIMERR ontology network. In the case of the KPI ontology the connection is done between the `kpi:Project` and the `io:InformationObject` classes representing the need to keep track of all the information generated during the renovation project. Examples of documents attached at this level are renovation plans, contracts, etc. In the case of the annotations ontology, the need to store files is to support the description of issues and problem, detected during the renovation activities, with complementary information such as images, audios, or videos. Finally, the building module also allows the attachment of files to the topological elements of the building. For instance, the connection of OBJ or IFC files per storey.

3.4.5 Model Particularities

This model does not present any particularity.

3.4.6 List of Resources

The following list of resources are provided for the information objects ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/information-objects/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-information-objects
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/information-objects/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/io_enriched.ttl

3.5 Material Properties Module

The Material Properties Ontology aims to provide the vocabulary to describe the building elements, their construction, their materials, and their corresponding properties.

3.5.1 Model description

The Figure 22 presents an overview of the classes, relations and attributes included in the Material Properties ontology. The purpose of the ontology is the description of building components defined within the BIMERR project that follow the material set definitions stated in the IFC 4 standard. Examples of these elements include walls, slabs and roofs. It is important to mention that building elements within the project that have not been defined using this convention, such as doors or windows, are not considered in this model, and can be directly described using properties sets in the building ontology. The material set definition indicates that an element may have a construction of type `mat:MaterialLayerSet`, `mat:MaterialProfileSet`, or `mat:MaterialConstituentSet`. These construction types can be further decomposed into `mat:MaterialLayer`, `mat:MaterialProfile`, and `mat:MaterialConstituent` respectively, where each of these entities has a set of related materials.

The materials can be described by a set of properties. For this purpose, we reuse the artifacts provided by the SAREF ontology. As we can see on Figure 22, the `mat:Material` class is related to the concept `mat:PropertySet`, which group together similar properties or properties that have a common denominator such was "properties for a wall". The properties inside a and their specific measurements by means of the relations `mat:hasProperty`, and `mat:hasMeasurement`, respectively. We avoid reusing the same relationships from the SAREF ontology because they have as domain the class `saref:FeatureOfInterest`, making `mat:PropertySet` a subclass of this concept, which is not semantically correct for our case.

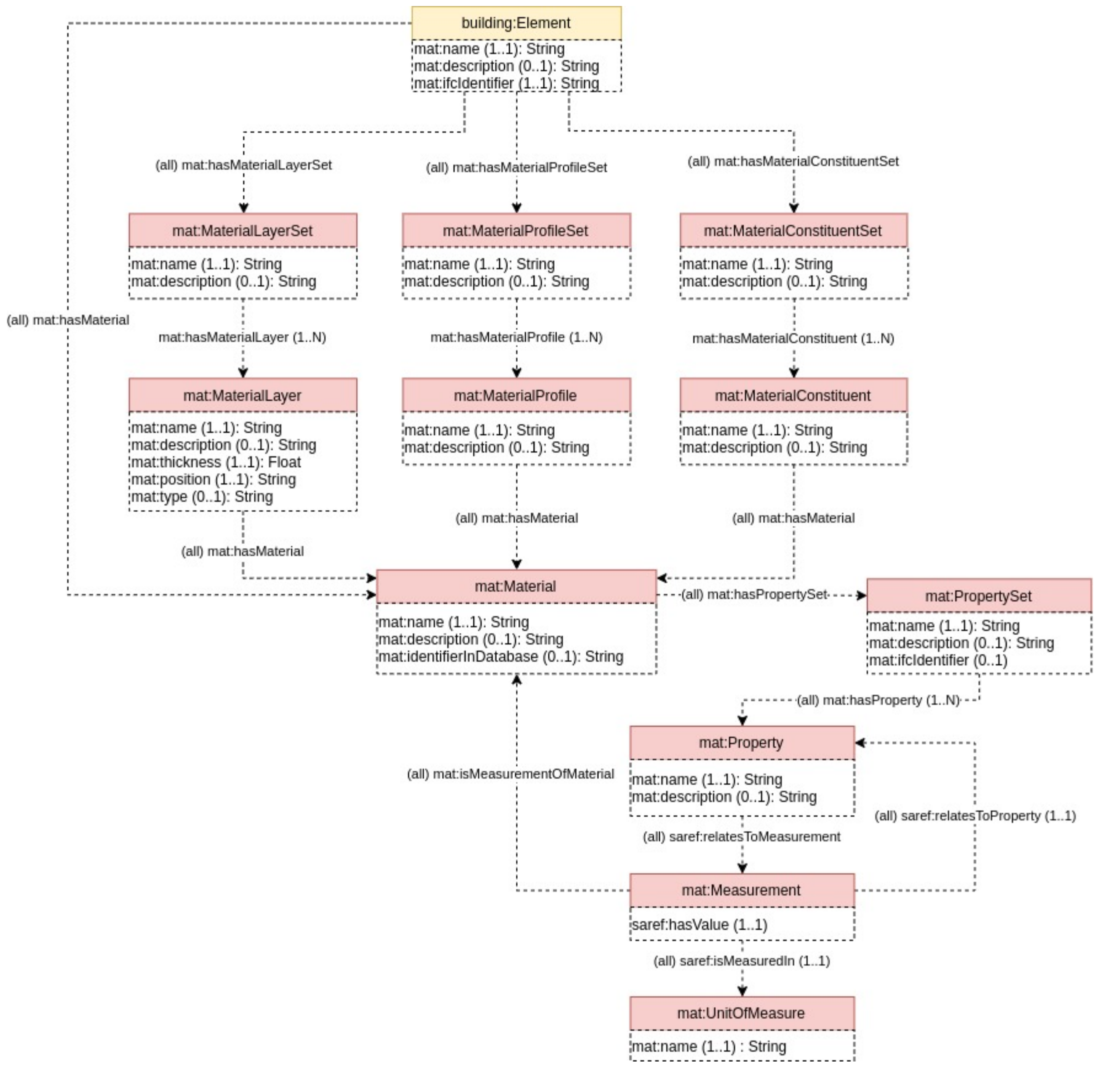


Figure 22. Materials Properties Ontology Conceptualization

3.5.2 Data Example

The Figure 23 shows an example of how we can use this ontology to describe the materials of a wall. The element described in this example is the wall data:Wall001 with IFC identifier "Wall01IFCGuid" that has a construction represented by the concept data:MaterialLayerSetWall01. This wall is composed by several layers, one of them is the layer data:MaterialLayerWall01 that has a thickness of 10 centimeters and is in the second position within the structure of the wall, where the ordering starts from the layer that interfaces with the internal spaces of the building.

This layer was made using brick masonry (`data:BrickMasonry_01`) as the main material, which has a set of properties that are organized in property sets (`data:PropertySet01`). One of the properties considered in this set is the Density (`mat:Density`), which for this specific material has a value equal to 1922 Kg-per-m3.

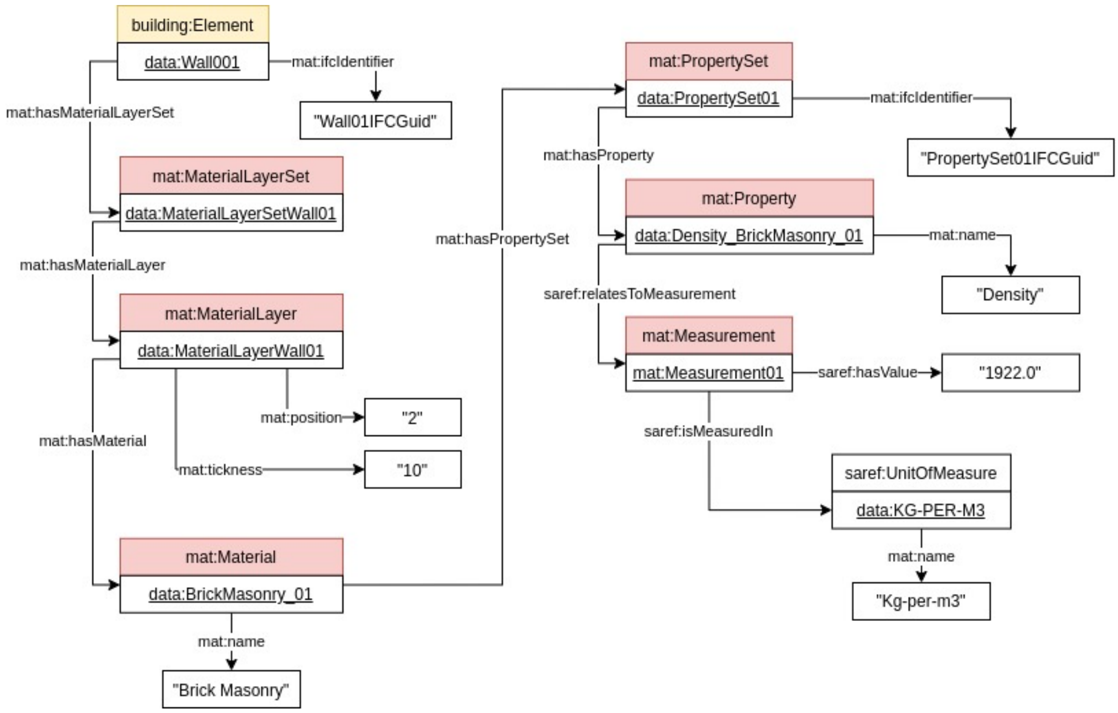


Figure 23. Data Example for the Material Properties Ontology

3.5.3 Model Evaluation

The model has been evaluated using two approached. The first one is related to the detection of pitfalls of the ontology using OOPS. The complete report can be seen in the ontology repository. The second method corresponds to the evaluation of the suitability of the ontology to represent the data it actually tries to model. This type of evaluation was performed during the development of the mappings between the data in the original format and the data in RDF format, which is finally stored in the Knowledge Graph Generator (KGG) component. For the case of the materials data model no problems arose during the transformation of the EnergyPlus materials database to RDF.

3.5.4 Links to other models

The Material Properties ontology is connected to the Building model through the `building:Element` concept. This link allows the description of the internal construction of building elements into more specific materials.

3.5.5 Model Particularities

This model does not present any particularity.

3.5.6 List of Resources

The following list of resources are provided for the material properties ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/material-properties/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-material-properties
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/material-properties/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/mat_enriched.ttl

3.6 OCCUPANCY PROFILE MODULE

The Occupancy Profile ontology has been developed to represent people's behavior inside buildings with a focus on the energy impact their actions produce. This information is critical to get more precise building energy profiles, and generate simulation scenarios that fit better with reality. In this sense, the shared conceptualization represented in this ontology focuses on the interaction of occupants with the different systems inside building spaces that could influence their thermal comfort requirements. This ontology has been developed using the Occupancy Behavior XML Schema [Hong et. al., 2019] as a basis, which defines the main. Some additional concepts as well as relations and attributes were added in order to obtain a full semantic representation of the domain.

The current conceptual model defined by the Occupancy Profile ontology is depicted in Figure 24. This model reuses some concepts and properties from the SAREF For Building Ontology (SAREF4BLDG), an extension of the SAREF ontology, that was created mainly to support interoperability between smart appliances in a building context. Some of the most important concepts related to the Occupancy Profile domain are the following:

- **Physical Object:** This concept refers to any object that has a proper space region such as the equipment or mechanisms within the building with which occupants may interact to restore or maintain environmental comfort.
- **Occupant:** This concept represents people that reside within a building or make use of specific spaces within it.
- **Behavior:** This concept is used to define the behaviors that people have inside a space. These behaviors could lead to interactions with systems.
- **Driver:** This entity represents the stimulating factors that provoke energy- related occupant behavior. A driver prompts a building occupant to perform either an action or in-action with a building system, impacting the energy use of a building.
- **Action:** This concept represents the interactions with systems or activities that occupants can perform to achieve environmental comfort.

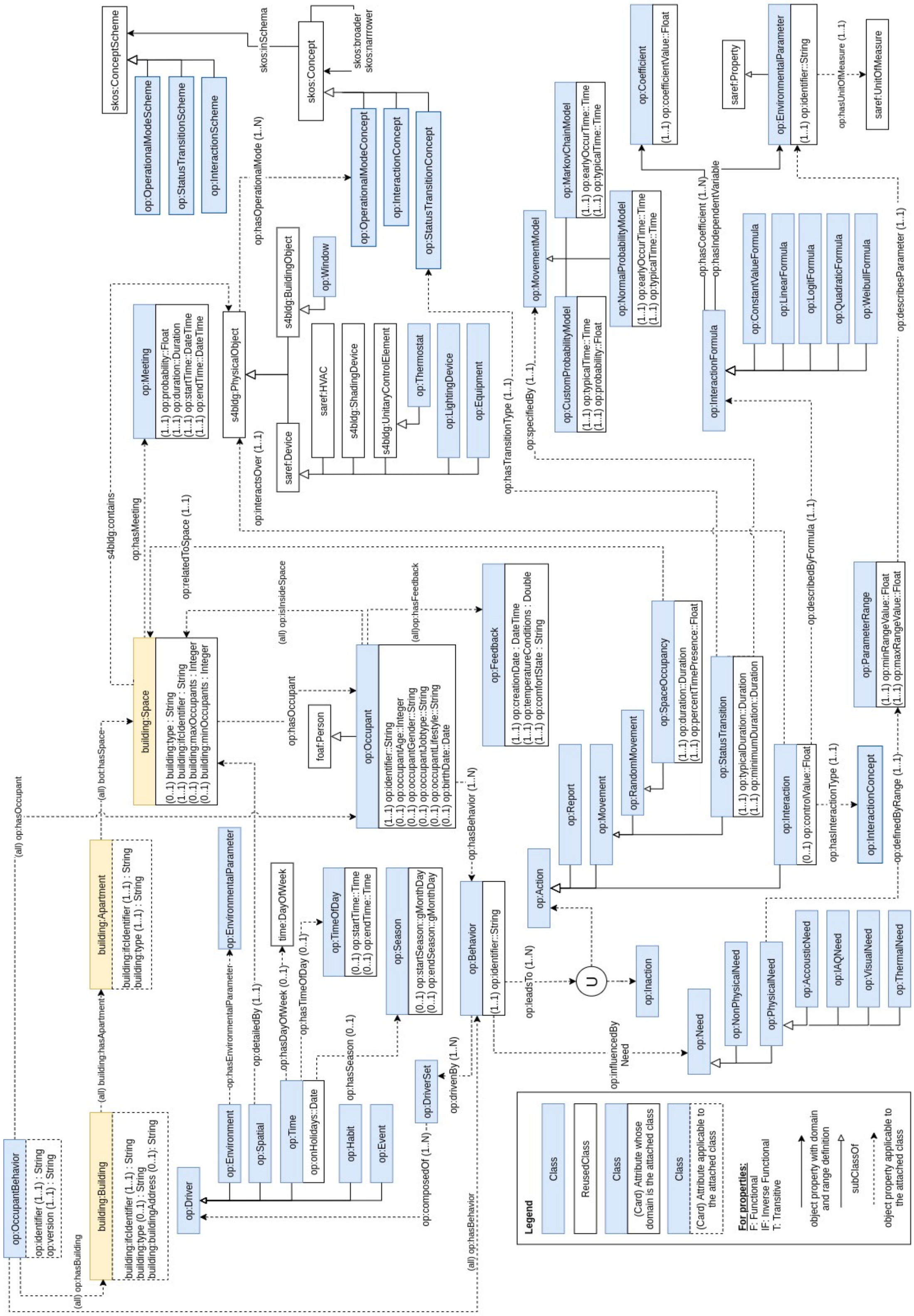


Figure 24. Occupancy Profile Ontology Conceptualization

3.6.1 Model description

The main concepts defined in the ontology and shown in Figure 24, are: `building:Building`, `building:Apartment`, `building:Space`, `op:Occupant`, `op:Behavior`, `op:Driver`, `op:Action` and `s4bldg:PhysicalObject` previously defined in Section 3.1. In the model conceptualization, we can observe that the class `building:Building` can have several spaces, which are concatenated in the concept `building:Apartment`, and is related to them through the property `bot:Apartment`. Those apartments have several spaces (`building:Space`). `building:Space` object can contain several systems that an occupant can interact with. From the obXML schema we have a list of systems that can be accommodated under the class `s4bldg:PhysicalObject`. Equipment such as HVAC's or shading devices can be represented by the classes `saref:HVAC` or s4bldg:ShadingDevice respectively. For systems not available in S4BLDG we created new terms, like the concept `op:Window`. Furthermore, the model provides a set of possible control options for these space systems. This enumeration of control options, which includes terms like `op:OnOff` or `op:Fixed`, is represented by the class `op:OperationalModeConcept`, subclass of `skos:Concept`. The use of the SKOS model enables a richer expressivity, by allowing the relation of the instances by means of the properties `skos:broader`, `skos:narrower` and others.

Several persons can occupy a specific space, where each occupant can have more than one behavior. Attached to the occupant concept, information about its personal profile can be indicated by means of the attributes `op:occupantAge`, `op:occupantGender`, `op:name`, `op:occupantJobtype`, and `op:occupantLifestyle`.

A behavior could refer to an action, where the concept `op:Action` involves the following subclasses: `op:Report`, `op:Movement` and `op:Interaction`. A person can also decide to not perform any action if its comfort levels are not satisfied; this scenario is modeled by means of the class `op:Inaction`. The interaction of an occupant with a system can be represented using mathematical models. This is addressed using the property `op:describedByFormula` where the destination is the concept `op:InteractionFormula`, which groups several off-the-shelve mathematical functions.

One example of specialization is the concept `op:LinearFormula` which gathers all the linear models that have at least one parameter in its definition.

For a mathematical formula to be properly defined we need a list of its coefficients and independent parameters. In order to model that, the relationships `op:hasCoefficient` and `op:hasIndependentVariable` were created. The first property (`op:hasCoefficient`) allows the connection from a given formula to an instance that will be linked to a certain numerical value and name by means of the data properties `op:coefficientValue` and `op:name`, respectively. The second property (`op:hasIndependentVariable`) links a given formula with a parameter that describes a certain environmental factor of the building space.

Additionally, the model should allow to express if the occupant needs are influencing in a certain way its behavior. For that purpose, the property `op:influencedByNeed` was implemented. These needs can be further categorized into Physical Needs, and Non-Physical Needs. Non-physical needs gather together all the possible instances that could not be quantified such as privacy, status, etc. On the other hand, Physical Needs represent comfort requirements that could be defined by fixed limits. To handle this, the concept `op:ParameterRange` was created, which indicates the maximum and minimum value for a specific environmental parameter (`op:EnvironmentalParameter`).

Apart from human-system interactions, there are other kind of occupant activities that are important from the energy consumption perspective. Occupant's movements (`op:Movement`), whether they occur at space or building level, can have an impact over the building energy profile. There are two types of movements covered in the model, random movements inside spaces, represented by the class `op:RandomMovement`, and status transition movements between spaces, represented by the concept `op:StatusTransition`. Status transition events can be approximated by probabilistic models such as the Markov Model, being the connection between both concepts the property `op:specifiedBy`. The particular attributes needed for each stochastic model are attached to their respective class. For instance the `op:NormalProbabilisticModel` requires the attributes `op:typicalTime` and `op:earlyOccurTime` during its instantiation.

Finally, a behavior can be caused by several factors, either internal or external. The object property `op:drivenBy` represents this relation, where the destination of the property is the class `op:DriverSet`, which allows to combine multiple drivers. This model decision allows to represent time (`op:Time`) as a driver by itself or as complementary information to describe when other type of drivers typically occur.

3.6.2 Data Example

Figure 25 shows a small example of how to use this model to represent occupancy behavioral data. The space `data:S2_Researcher_Office` is an office occupied by 4 researchers. In this example only 2 occupants are described. This space contains two energy related systems: A thermostat (`data:Thermostat_1`) and a HVAC (`data:HVAC_1`). The first occupant is the `data:OC_Researcher_1` researcher which has the behavior to set the thermostat to 23ºC according to a certain probability. The second researcher (`data:OC_Researcher_2`) has the behavior to set the HVAC system to 27ªC. Additionally, it is possible to indicate the operational mode of the HVAC that is this case is zone on off (`data:ZoneOnOff`).

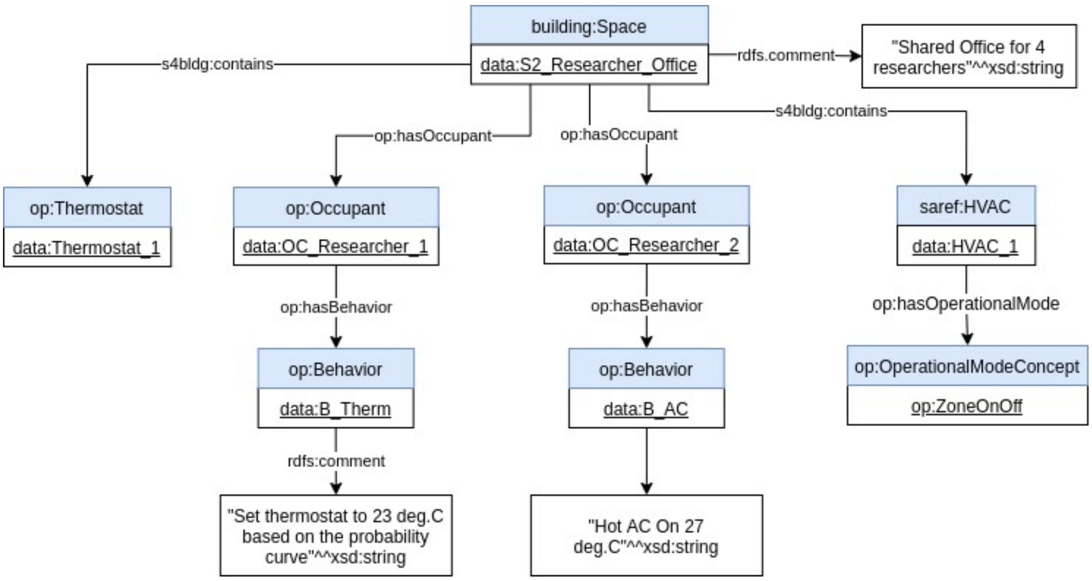


Figure 25. Data Example for the Occupancy Profile Ontology

3.6.3 Model Evaluation

The model has been evaluated using two approached. The first one is related to the detection of pitfalls of the ontology using OOPS!. The complete report can be seen in the ontology repository.

The second method evaluates the suitability of the model to the BIMERR Interoperability Framework. After the evaluation, additional concepts and relations were included in order to support the all functionalities of BIF. For instance, in order to facilitate the mapping and the access of the data inside the model, it was necessary to create the `op:OccupantBehavior` concept that acts as a root node from which the rest of concepts can be accessed. Furthermore, additional inverse relationships were added such as the `op:isInsideSpace` object property between the `op:Occupant` and `building:Space` concepts.

3.6.4 Links to other models

The occupancy profile ontology is highly linked to the building ontology by means of the `building:Building`, the `building:Apartment`, and the `building:Space` concepts which provide the context to being able to describe the behaviors of the residents.

3.6.5 Model Particularities

In order to allow the appropriate communication between the PRUBS tool and the BIF, the ontology includes the concept `op:OccupantBehavior` as a root node. Semantically this node is not needed in the ontology, however, it allows a smooth mapping between the obXML format and the BIF data model.

3.6.6 List of Resources

The following list of resources are provided for the occupancy profile ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/occupancy-profile/
- **Ontology GitHub repository:**

https://github.com/oeg-upm/bimerr-occupancy-profile

- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/occupancy-profile/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/op_enriched.ttl

3.7 Sensor Data Module

The Sensor Data ontology has been developed to represent measurements generated by sensors located inside building spaces. These devices will be used to estimate building resident energy profiles, by measuring indirectly the interaction between occupants and systems. For the construction of this ontology, the SenML standard [SenML, 2018] was taken as a basis.

The current conceptual model defined by the Sensor Data ontology is depicted in Figure 26. This model reuses concepts and properties from the SAREF ontology which is intended to enable interoperability between different IoT solutions from different manufacturers. Some of the most important concepts taken from SAREF are the following:

- **Property:** An aspect of an entity that can be observable by a sensor.
- **Device:** A tangible object designed to accomplish a particular task in households, common public buildings or offices. In order to accomplish this task, the device performs one or more functions.
- **Measurement:** Represents the measured value made over a property. It is also linked to the unit of measure in which the value is expressed and the timestamp of the measurement.
- **Unit of Measure:** The unit of measure is a standard for measurement of a quantity, such as a Property.

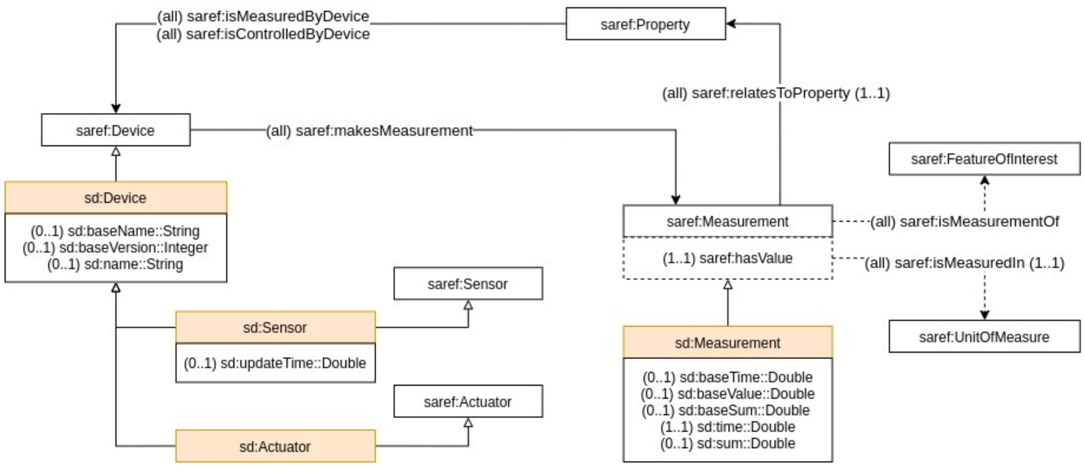


Figure 26. Sensor Data Ontology Conceptualization

3.7.1 Model Description

As mentioned previously, the sensor data module has been constructed with the idea of being able to represent data coming from sensors or, if necessary, being able to model commands sent to system actuators. Following this design principle, the `saref:Device` concept was used to represent all the individuals that will perform those functions inside building spaces. According to the SenML standard, a device is related to a record, which could be either a measurement or a command. This relation is modeled by the object property `saref:makesMeasurement` which has as destiny the class `saref:Measurement`, that groups individual records. The attribute `saref:hasValue` indicates quantitatively the magnitude of the record.

Even though most of the IoT streams coming from the Middleware module in the BIMERR platform will be resolved (i.e. preprocessed), it is recommended to include most of the fields provided by SenML in the model. This may be useful for making more complex queries. In order to model these extra attributes, we need to extend the representation capabilities of `saref:Device` by means of the concept `sd:Device`. One of the most relevant attributes attached to this entity is `sd:sum` which represents the integrated sum of record values over time. This is useful to notice when a sensor has lost communication with the server. Other elements incorporated are the base versions of the attributes described previously such as `sd:baseValue`, that with `saref:hasValue` property are added together to get the value of the measurement.

Also, we need to relate the measurement to a specific property of the environment or aspect of an entity. Through the use of the object property `saref:relatesToProperty`, we can model this connection where the range of the relation is the concept `saref:Property`, that groups individuals such as temperature, illuminance, or noise level. For a full description of a record we need to indicate the units in which it is expressed. The concept `saref:UnitOfMeasure` takes responsibility for that function further detailing the concept `saref:Measurement` throughout the relation `saref:isMeasuredIn`.

Finally, a record is not only an estimation of a physical property of the world, it also describes a particular thing or context as for instance, the temperature or illuminance within an apartment's room. With this in mind, the property `saref:isMeasurementOf` is used to link a measurement to the concept `saref:FeatureOfInterest`, which represents all the entities that are being described or acted over by the devices.

3.7.2 Data Example

Figure 27 shows an example of how to use this model to represent sensor information and measurements related to the BIMERR project. The `data:Sensor01` sensor is an occupancy sensor that made one measurement (`saref:makesMeasurement`) of 80% of occupancy (`data:Occupancy`) at an specific space.

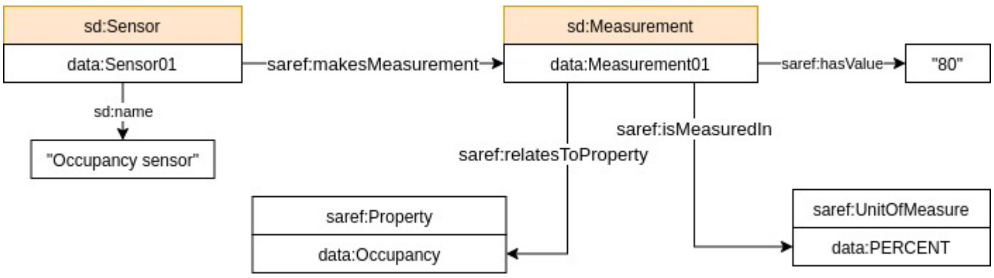


Figure 27. Data Example for the Sensor Data Ontology

3.7.3 Model Evaluation

The model has been evaluated using OOPS!, the detailed results are available at the ontology repository.

3.7.4 Links to other models

The ontology is connected to the network by means of the `saref:Device` concept, which is a subclass of the `s4bldg:PhysicalObject` concept in the occupancy profile ontology.

3.7.5 Model Particularities

This data model does not has particularities.

3.7.6 List of Resources

The following list of resources are provided for the sensor data ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/sensor-data/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-sensor-data
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/sensor-data/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/sd_enriched.ttl

3.8 Key Performance Indicator Module

The Key Performance Indicator module aims to model the metrics defined at the beginning of building renovation activities to monitor the conformance with typical requirements related to energy efficient buildings. This type of measures serves not only to get a correct estimation of the advances in the renovation workflow, but also to adjust the planning of the construction activities and improve the communication with stakeholders. The conceptualization proposed also focuses on the relation between KPIs, projects and scenarios. The construction of this ontology has been done using the requirements and needs identified for the PWMA and RenoDSS BIMERR applications.

The up-to-date conceptual model defined for the Key Performance Indicator ontology is illustrated in Figure 24. This model follows good ontology engineering practices, and reuses some terms from existing models, specifically concepts from SAREF for City (SAREF4CITY), which is an extension of the SAREF ontology. Some of the most relevant concepts in the model are defined below:

- **KPI:** A type of performance measurement. KPIs evaluate the success of an organization or of a particular activity in which it engages.
- **KPI Value:** A quantity assessment of the KPI being evaluated. It also takes into account the time at which the indicator was calculated.
- **Project:** The building renovation project the KPI metric is related to.
- **Scenario:** The particular conditions under which the KPI is being evaluated.

3.8.1 Model Description

The main idea behind this module is to represent some concepts related to the monitoring of project processes and also check if the energy performance indicators established for renovation projects are being satisfied. With this goal in mind, the whole conceptualization circulates around the term `kpi:KPIValue`, that as defined previously in section 3.3, is a detailed specification of a performance indicator. This new concept is an extension of the class `s4city:KeyPerformanceIndicatorAssessment` of SAREF4CITY ontology, which already includes some attributes useful for our purposes such as `saref:hasValue`. Those attributes allow us to state a numerical value to the metric being evaluated as well as a detailed description of what is being assessed. KPI values define quantitatively a specific indicator. This relation is modeled by the object property `s4city:quantifiesKPI`, that connects to the `kpi:KPI` concept. In order to reuse some of the attributes and relations previously defined in the SAREF4CITY ontology, we model the `kpi:KPI` concept as a subclass of the entity `s4city:KeyPerformanceIndicator`. It is possible to use the attribute `kpi:name` to provide a proper name to our KPI individuals. However, the units in which we are expressing the value of performance indicators also need to be provided, we model that by adding the attribute `kpi:unit` to the KPI definition. Additionally, data properties such as `kpi:identifier` and `kpi:description` are added to achieve a full description of the concept.

Finally, a KPI value exists within the context of a project, which is the activity we are trying to measure the success of. This relation is represented by the property `kpi:relatedProject` linking a KPI value to the concept `kpi:Project`. Furthermore, a KPI value is calculated under certain conditions. The conceptualization models this set of conditions by means of the class `kpi:Scenario`. The connection used to link both concepts is the object property `kpi:relatedScenario`. Each scenario has several renovation measures that are applied over specific elements of the building.

3.8.2 Data Example

Figure 29 illustrates an example on how we can use this ontology to describe KPI's. In this case, the KPI to describe is `data:ScaffoldCost` which is related to the scaffold installation cost. This KPI is quantified by the data:ScaffoldKPIValue01 entity, which has a value of 300000 euros. This KPI value is related to the `data:BimerrProject01` project and the `data:SimulatedScenario001` scenario. At the same time, this KPI value can be estimated given a set of parameters such as time, cost per unit, and number of units of scaffold.

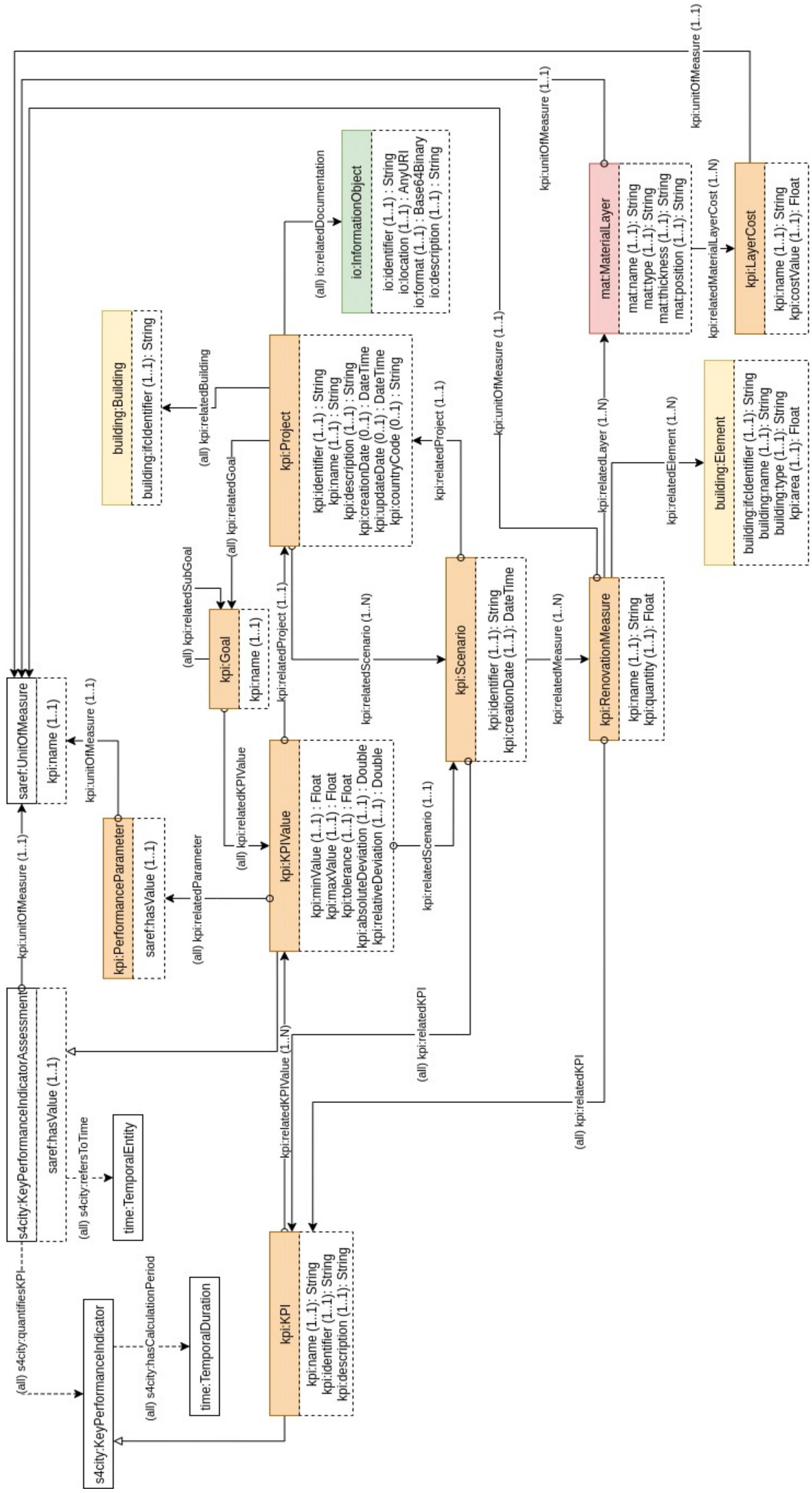


Figure 28. Key Performance Indicator Ontology Conceptualization

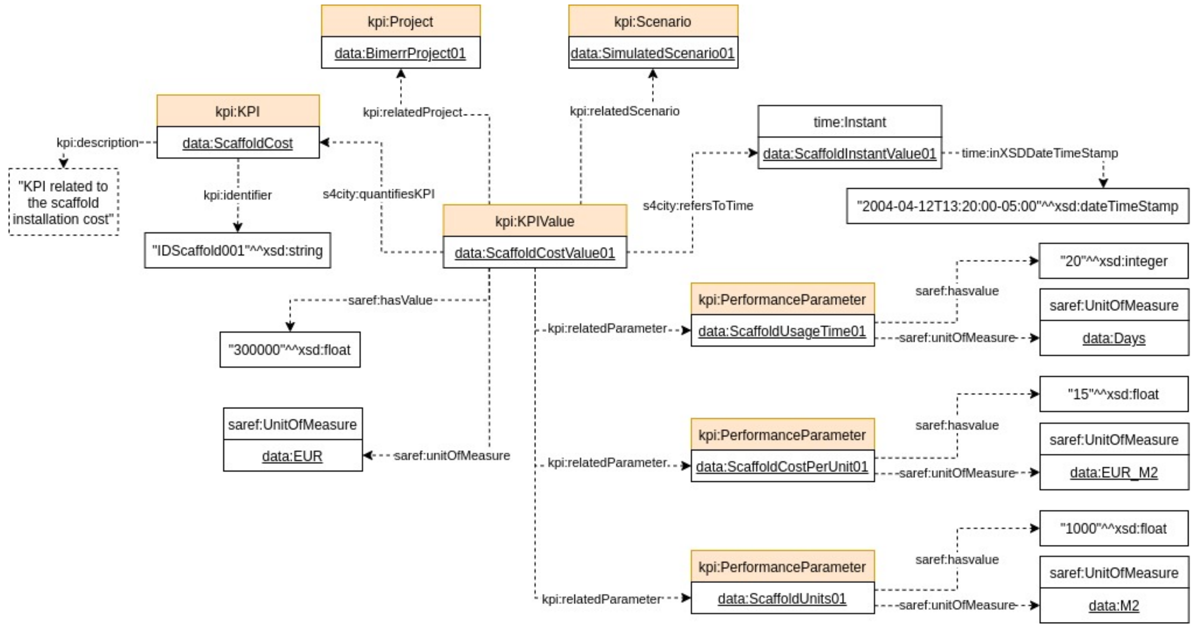


Figure 29. Data Example for the Key Performance Indicators Ontology

3.8.3 Model Evaluation

The model has been evaluated using two approached. The first one is related to the detection of pitfalls of the ontology using OOPS. The complete report can be seen in the ontology repository.

The second method evaluates the suitability of the model to the BIMERR Interoperability Framework. After the evaluation, additional concepts and relations were included or modified in order to support the full functionalities from BIF. For instance, `kpi:name` and `kpi:description` attributes were added to the `kpi:Project` concept. Furthermore, the change of the label of the `kpi:ISO3166` attribute to `kpi:countryCode`. Finally, all the relationships between two concepts should include in their label the prefix related, e.g. the `kpi:affectedELement` relation was changed by `kpi:relatedElement` object property.

3.8.4 Links to other models

The KPI model is linked to several other domains. First, the model is connected to the `io:InformationObject` concept in order to attach documents at the Project level. Another connection is with the building module, where the objective is to indicate to which building element the specified renovation measure applies. Furthermore, there is another relationship with the building module, between the `kpi:Project` and the `building:Building` concepts, to facilitate the querying of all the buildings involved in a renovation project. Finally the connection with the material properties data model by means of the `mat:MaterialLayer` occurs, in order to indicate that certain measures only apply to specific layers of certain building elements, such as walls.

3.8.5 Model Particularities

This data model does not have particularities.

3.8.6 List of Resources

The following list of resources are provided for the key performance indicator ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/key-performance-indicator/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-key-performance-indicator
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/key-performance-indicator/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/kpi_enriched.ttl

3.9 WEATHER MODULE

The Weather ontology has been developed to define climate properties and locate their measurements. This module is intrinsically linked to the sensor data ontology because most of the data used in this domain consist of measurements.

The current conceptual model defined by the Weather ontology is depicted in Figure 30. This model reuses concepts and properties from the SAREF, SAREF for City (SAREF4CITY) and WGS84 Geo Positioning (WGS84_POS) ontologies. The first two ontologies have been discussed and reused on the previous domains. The last model (WGS84_POS), is an ontology that provides the Semantic Web community with a namespace for representing latitude, longitude and other information about spatially-located things, using WGS84 as a reference datum.

Some of the most important concepts taken from SAREF4CITY are the following:

- **Administrative Area**: An administrative division, unit, entity, area or region, also referred to as a subnational entity, constituent unit, or country subdivision, is a portion of a country or other region delineated for the purpose of administration.
- **City**: A city is a large human settlement. A city is distinguished from other human settlements by its relatively great size, but also by its functions and its special symbolic status, which may be conferred by a central authority.
- **Country**: A country is a region that is identified as a distinct national entity in political geography.

Also, some of the most important concepts taken from WGS84_POS are the following:

- **SpatialThing**: Anything with spatial extent, i.e. size, shape, or position, e.g. people, places, bowling balls, as well as abstract areas like cubes.
- **Point**: A point, typically described using a coordinate system relative to Earth, such as WGS84.

3.9.1 Model Description

The main concept defined in the ontology, as shown in Figure 30 is `weat:WeatherProperty`. In this model conceptualization we can observe that almost all the model is created by reusing SAREF classes as we mentioned earlier, such as `saref:Device`, `saref:Property`, `saref:FeatureOfInterest`, etc. We used this ontology as reference because the model is applicable to the data needs, it is a standard ontology proposed by the European Telecommunications Standards Institute1 and there is a family of related ontologies that can be reused for other BIMERR ontology modules.

As it can be seen, the observations or measurements (`saref:Measurement`) are related to the observed property (`saref:Property`), the related feature of interest (`saref:FeatureOfInterest`), and the unit of measurement (`saref:UnitOfMeasure`) in which the value is provided and the time stamp in which the measure was taken. The `weat:WeatherProperty` is defined as a subclass of `saref:Property` and the weather properties about which there will be measurements in the data are represented as instances of `weat:WeatherProperty` as shown in Figure 31. Instances of `saref:UnitOfMeasure` are depicted in Figure 32. The values of the specific measurements would be attached to each instance of `saref:Measurement` by means of the property `saref:hasvalue`.

On the other hand, a `saref:Device` location can be indicated by the property `geo:location` from the WGS84_POS vocabulary.

SAREF4CITY is used for the class `s4city:AdministrativeArea` with its subclasses `s4city:City` and `s4city:Country`, to locate this `saref:Device` at a country and city.

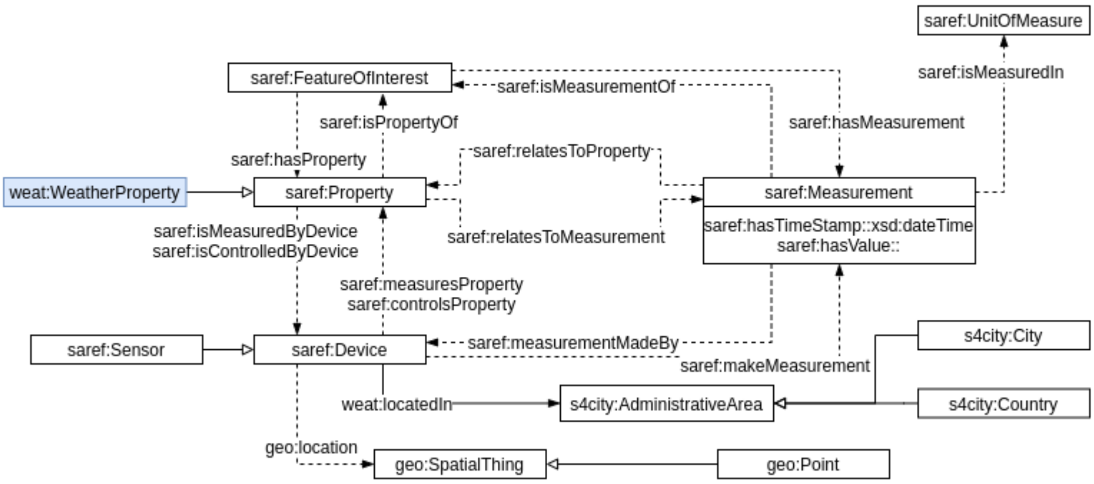


Figure 30. Weather ontology conceptualization

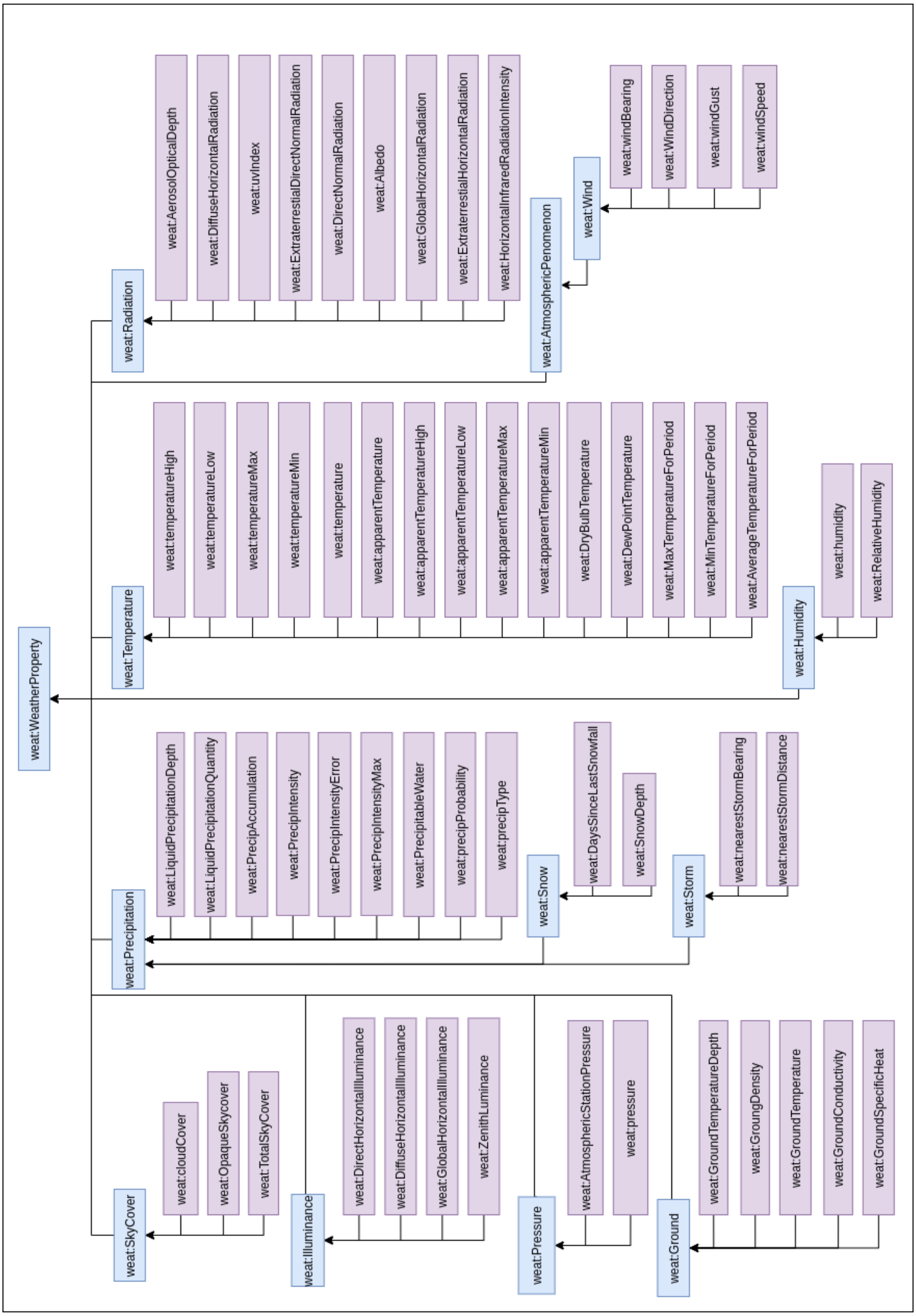


Figure 31. Weather ontology individuals.

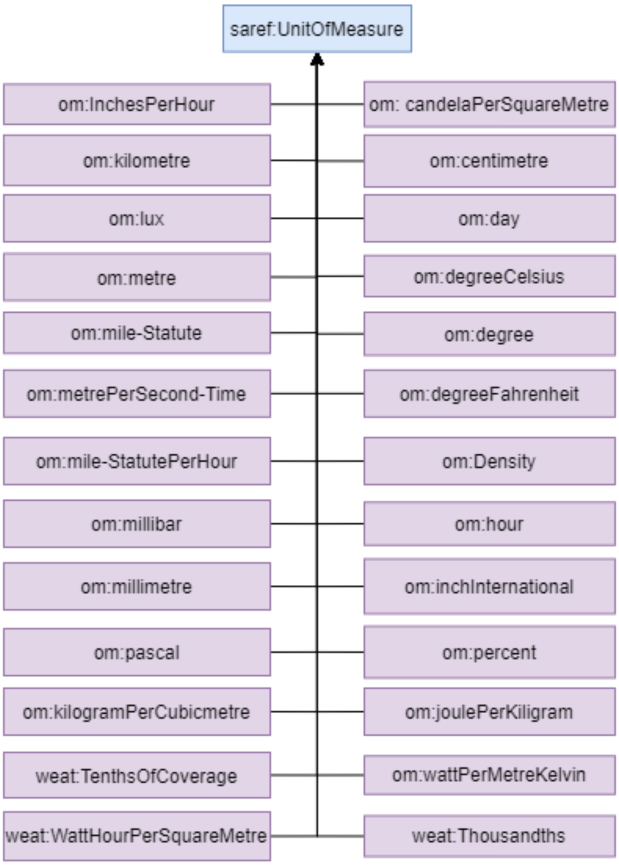


Figure 32. Units of measure individuals.

3.9.2 Data Example

Figure 33 shows an example of how to instantiate the ontology. The `data:Sensor01` sensor is located in the city of Madrid. The property this sensor measures is the temperature (`weat:Temp`). One of the records taken by the sensor is the `data:Measurement_Temp01` measurement which has a value 35 degrees.

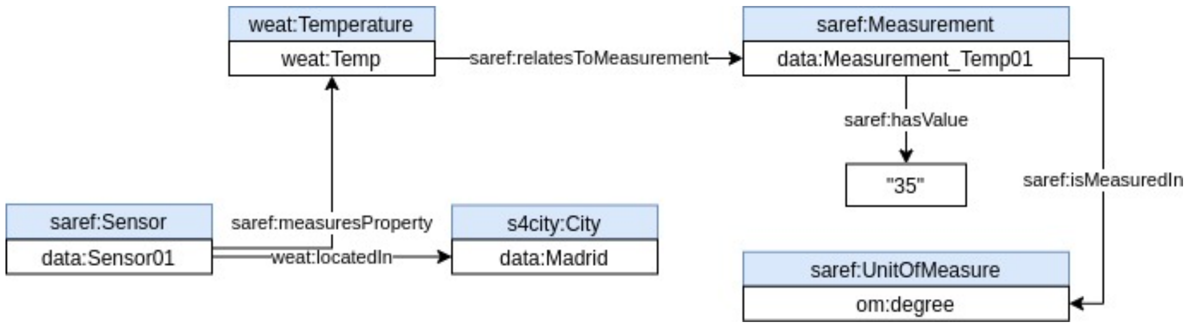


Figure 33. Data Example for the Weather Ontology

3.9.3 Model Evaluation

The model has been evaluated using two approaches. The first one is related to the detection of pitfalls of the ontology using OOPS! . The complete report can be seen in the ontology repository. The second method corresponds to the evaluation of the suitability of the ontology to represent the data that it actually tries to model. This type of evaluation was performed during the development of the mappings between the data in the original format and the data in RDF format, which is finally stored in the Knowledge Graph Generator (KGG) component. For the case of the weather ontology no problems appeared during the transformation of the EnergyPlus Weather and Open Weather Map databases to RDF.

3.9.4 Links to other models

The ontology is connected to the network by means of the `saref:Device` concept, which is a subclass of the `s4bldg:PhysicalObject` concept in the occupancy profile ontology.

3.9.5 Model Particularities

This model does not have any particularity.

3.9.6 List of Resources

The following list of resources are provided for the weather ontology:

- **Ontology URI:**
  https://bimerr.iot.linkeddata.es/def/weather/

- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-weather
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/weather/ontology.ttl
- **Ontology enriched with metadata:** https://bimerr.iot.linkeddata.es/ontologies/weat_enriched.ttl

4 BIMERR Ontologies Transformation to Data Models

This section describes a summary of the transformation pipeline developed to convert the different ontologies discussed in the previous sections into the data model that is finally utilized by the BIMERR Interoperability Framework. For a more detailed explanation about the converter functionality, please refer to the deliverable D4.5.

4.1 Metadata Ontology

The metadata ontology is a complementary model that provides the vocabulary needed to annotate with extra information ontology elements from any module of the BIMERR ontology network. The annotation of the ontologies with these terms is the first step in the transformation process needed to generate the JSON data model required by BIF. Table 5 gives a complete list of the vocabulary included in the model, the equivalent fields in the JSON data model after the transformation, and a definition for each term.

Table 5. Metadata list

Metadata on the Ontology	Metadata on Data Model	Description
rdfs: comment	Definition	A brief overview that acts as an account of a concept's contents
rdfs: label	Related_terms	A set of related terms (e.g. synonyms) that can be alternatively used to represent the concept
skos:altLabel	Related_terms	A set of related terms (e.g. synonyms) that can be alternatively used to represent the concept
bm:isDefinedByStandard	standards	A list of standards in which the specific concept is modeled, along with their complementary information, i.e. the exact concept used in such a standard, its type (e.g. element, attribute) and its use (required/optional) that applies for attributes
dc: created	Date_added	The date when the specific concept was added in the

		data model
bm: deprecated	Date_deprecated	The date when the specific concept became obsolete in the data model
owl: versionInfo	Version	The version of the model when the concept was last modified
bm: ordered	Ordered	An indication of whether ordering is needed for multiple appearances of the same concept.
bm: sensitive	Sensitive	An indication whether the specific concept models personal or sensitive data
bm: transformation	transformation	Information for the transformation rules that are related to a specific applicable standard. It practically contains the function that is required for the transformation and the parameters / concepts that are involved.
bm: measurementType	measurementType	An indication of the measurement type that is applicable to a concept, e.g. referring to distance, temperature, etc.
bm: measurementUnit	measurementUnit	The baseline measurement unit for the specific concept and measurement type
bm: timeZone	timeZone	The timezone to which the data refer by default.
bm: codeList	codeList	A link to the code list that should be typically used for the data that refer to the specific concept
bm: codeType	codeType	The type of code list that is applied for the specific concept

Some of the terms in this model were extracted from well-known vocabularies; for instance, the `dc:created` term from the Dublin Core metadata initiative or the `skos:altLabel` term from the SKOS Core vocabulary. However, there are other terms that were created in order to cover the requirements from the BIF side. For instance, the inclusion of the term `bm:sensitive` to model sensitive data or `bm:timeZone` to indicate the time zone of the data. In Table 5 we can identify those new terms by the prefix bm which corresponds with the namespace http://bimerr.iot.linkeddata.es/def/bimerr-metadata.

The output of the annotation process are enriched versions of the ontologies that are stored as different files in the BIMERR ontology portal. The enriched models can also be directly accessed by the links provided in the List of Resources sections of each ontology description in the present document.

4.1.1 List of Resources

The following list of resources are provided for the metadata ontology:

- **Ontology URI:** https://bimerr.iot.linkeddata.es/def/metadata/
- **Ontology GitHub repository:** https://github.com/oeg-upm/bimerr-metadata
- **Ontology in Turtle format:** https://bimerr.iot.linkeddata.es/def/metadata/ontology.ttl

4.2 TRANSFORMATION PROCESS

The BIMERR Ontology to Data Model converter is a web service, which given an ontology implemented in OWL performs a series of transformations to finally return a JSON serialized data model. This converter should receive an enriched version of the original ontology in order to extract all the metadata required by BIF platform.

The complete transformation pipeline is depicted on Figure 34. The process starts with the enrichment step using an ontology editor, where the ontology engineer instantiate the extra fields required to generate the enriched version of the ontology. The original ontology is not edited in this process, but the changes are stored in a separated file. The BO2DM service takes as input this enriched ontology and triggers a series of conversion steps to finally generate the data model. Any changes produced to the original ontology will propagate to the enriched one that at the same time will materialize those changes in the data model through the BO2DM converter. Finally, once the data model is obtained it is send to BIF for its implementation. In case some modifications are needed from the BIF user sides in the model, the changes are communicated to the ontology development team members, who update the ontology or the enriched version depending on the type of change. For more details with respect to the conversion rules, assumptions, and restriction of this component, the reader should review the final version of deliverable D4.5 BIMERR Building Semantic Modelling tool 2.

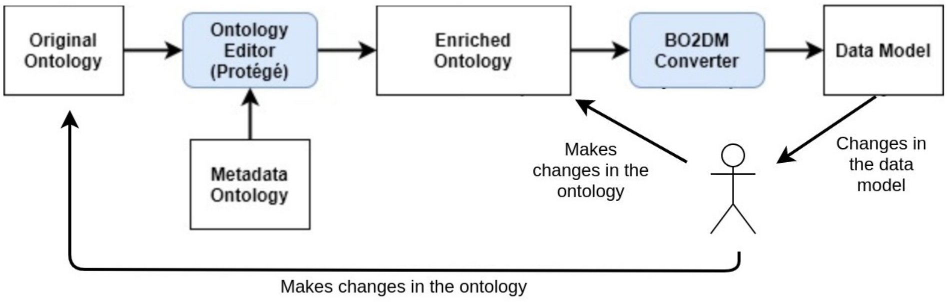


Figure 34. Ontology to Data Model Conversion Pipeline

The Figure 35 shows an example of the JSON data model finally generated by the BO2DM converter from the Information Objects ontology.


```
{
  "InformationObject": {
    "definition": "Class that represents any type of file.",
    "related_terms": [
      null
    ],
    "standards": [
      null
    ],
    "date_added": null,
    "date_deprecated": null,
    "version": "0.0.1",
    "children": {
      "datatype": {
        "definition": null,
        "related_terms": [
          null
        ],
        "standards": [
          null
        ],
        "date_added": null,
        "date_deprecated": null,
        "version": "0.0.3",
        "facet": {
          "cardinalityMax": "1",
          "ordered": null,
          "sensitive": null,
          "transformation": null,
          "measurementType": null,
          "measurementUnit": null,
          "timeZone": null,
          "codeList": null,
          "codeType": null
        },
        "type": "base64Binary"
      },
```


Figure 35. Information Objects data model

CONCLUSIONS

This document has presented the final state of the BIMERR ontology network and the data models developed within the context of the T4.2 activities. The BIMERR ontology network is composed of 9 modules, each representing a domain that has been identified within BIMERR. From those modules, 7 support the exchange of heterogeneous data between the BIMERR applications, while 2 support the publication of some publicly accessible databases in RDF format in the Knowledge Graph Generator. It is important to mention that even though the final state of the BIMERR ontology covers the whole set of requirements from the tools and use cases, new requirements can appear that may introduce additional modifications to the data model. Additionally, a complementary metadata ontology has been developed to support the data model generation that is required by BIF.

During the processes and different tasks activities related to the BIMERR Interoperability Framework it became apparent that the well-known Competency Questions technique [Grüninger and Fox, 1995] is sometimes not the most suitable solution for every project and situation. Although it might be helpful to identify main concepts and top-level relations, it is not optimal for all use cases. For example, when the interlocutor's profile is closer to the actual data to be modelled in the ontology or data model, more structure artefacts help to better describe the data exchange requirements. In this sense, the use of adapted templates from METHONTOLOGY [Fernández-López et. al., 1997] [Fernández-López et. al., 1999] has greatly improved the efficiency and communication between ontology and data model developers and the partners involved in the data extraction requirements. In addition, such templates have been designed taking into account the ontology and data model needs, gathering in one place all the information. This feature allowed us to reduce the number of iterations and resources generated during the task, and therefore, the need to constantly maintain the templates. In other cases, mostly when the model development has already started, it has been proved more practical to use a conceptual model as a communication tool with domain experts so that new entities and relations are reflected directly in the shared models rather than in a requirements list.

Another methodological resource generated from this stage was Chowlk. A framework that accelerated the development of the conceptualizations and improves the understanding of the models created. It also allowed us to automate the generation of the first versions of the ontological implementations. The Chowlk framework is composed of two parts: a) the visual notation and b) the converter from the conceptual diagrams to the OWL code. Another output from the application of this tool was the refinement of the visual shapes included in the first version of the visual notation, which allows us to be more effective in the representation of the models.

The existence and management of two versions of the BIMERR model, the ontology version and the JSON data model version, imposed some restrictions and requirements that were needed to be taken into account when modeling the domains.

For the next steps, even though the T4.2 has concluded, the outputs and discussions from the BIF integration activities will be taken into account in case minor updates are needed on the ontology network. At this time, it is still needed to test the integration of data for the Annotation Objects, Information Objects, Renovation Processes, and Material Properties data models.

BIBLIOGRAPHY

[Chávez-Feria et. al., 2020] Chávez-Feria, S., Poveda-Villalón, M., Bosché, F., Bountouni, N., Fenz, S., García-Castro, R., Giannakis, G., González-Gerpe, S., Kollmer, S., Kousouris, S., Thanos, T., Lampathaki, F., Valero, E., Katsigarakis, K., Falcioni, D., Straka, M., Varga, J., Lilis, G. (2020). D4.2 BIMERR Ontology & Data Model 1, Technical Report, Universidad Politécnica de Madrid (UPM), 2020. BIMERR Project. https://bimerr.eu/.

[Fernández-López et. al., 1997] Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). METHONTOLOGY: from ontological art towards ontological engineering. In Proceedings of the Ontological Engineering AAAI97 Spring Symposium Series. American Association for Artificial Intelligence.

[Fernández-López et. al., 1999] Fernández-López, M., Gómez-Pérez, A., Pazos-Sierra, J., and Pazos-Sierra, A. (1999). Building a chemical ontology using methontology and the ontology design environment. IEEE Intelligent Systems, 14(1):37–46.

[García-Castro et. al., 2017] García-Castro, R., Fernández-Izquierdo, A., Heinz, C., Kostelnik, P., Poveda-Villalón, M. and Serena, F. (2017, August 31) Detailed Specification of the Semantic Model, Technical Report, Universidad Politécnica de Madrid (UPM), 2017, VICINITY Project. https://vicinity2020.eu.

[Garijo et. al., 2020] Garijo, Daniel. "WIDOCO: a wizard for documenting ontologies." (2017) International Semantic Web Conference. Springer, Cham.

[Grüninger and Fox, 1995] Grüninger, M. and Fox, M. (1995). Methodology for the design and evaluation of ontologies. In IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing.

[Hong et. al., 2019] Hong, T., D'Oca, S., Taylor-Lange, S. and Turner, W. (2015). An ontology to represent energy-related occupant behavior in buildings Part II: Implementation of the DNAS framework using an XML schema. http://dx.doi.org/10.1016/j.buildenv.2015.08.006

[Poveda-Villalón, 2012] Poveda-Villalón, M. A reuse-based lightweight method for developing linked data ontologies and vocabularies (2012, May). In *Extended Semantic Web Conference* (pp. 833-837). Springer, Berlin, Heidelberg. May 2012.

[Poveda-Villalón et. al., 2014] Poveda-Villalón, M., Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2014). OOPS! (ontology pitfall scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS), 10(2), 7-34.

[Poveda-Villalón et. al., 2018] Poveda-Villalón, M., García-Castro (2018). Extending the SAREF ontology for building devices and topology, Semantic Web Journal, IOS Press.

[Poveda-Villalón et. al., 2019b] Poveda-Villalón, M., Fernández-Izquierdo, A., and García-Castro, R. (2019, January 14). Linked Open Terms (LOT) Methodology (Version 1.0). Zenodo.

[Rasmussen et. al., 2020] Rasmussen, M., Lefrançois, M., Ferdinand, G., Pauwels, P. (2020). BOT: the Building Topology Ontology of the W3C Linked Building Data Group, Semantic Web Journal, IOS Press.

[Sabou et. al., 2012] Sabou, M., Fernandez, M. (2012). Ontology (network) evaluation. In Ontology engineering in a networked world (pp. 193-212). Springer Berlin Heidelberg.

[SenML, 2018] Jennings, C., Shelby, Z., Arkko, J. (2013) Media Types for Sensor Markup Language (SenML). https://tools.ietf.org/html/rfc8428 (Work in process, Last update: 08-2018)

[Tiwari et. al., 2019] Tiwari, S., Alexopoulou, E., Bosché, F., Bountouni, N., Fenz, S., García-Castro, R., Giannakis, G., Kakardakos, T., Koussouris, S., Kyprianidis, A., Neubauer, T., Ntalapera, D., Papapolyzos, T., Parna, E., Prekas, G., Ratajczak-Jeziorska, J., Rontogianni, E., Tsakiris, T., Vafeiadis, G., Valero, E., and Zacharis, E. (2019) Report on Semantic Alignment & Linking of EEB-related ontologies, Technical Report, Universidad Politécnica de Madrid (UPM), 2019. BIMERR Project. https://bimerr.eu/.