This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957047.

H2020-LC-SC3-EE-2020-1/LC-SC3-B4E-6-2020

Big data for buildings

BIGG

Building Information aGGregation, harmonization and analytics platform

Project Nº 957047

D4.2

Description of the final harmonization layer

Responsible:	Stoyan Danov (CIMNE)
Document Reference:	D4.2
Dissemination Level:	Public
Version:	Final
Date:	22/11/2023

Executive Summary

This document describes the development and the integral parts of the BIGG harmonization layer. The harmonization layer is a core component in the BIGG Reference Architecture Framework that ensures the interoperability with external data sources, defines the data structure of the internal databases, enables the integration between the different components, and provides a harmonised data input to the BIGG AI Toolbox.

The final harmonization layer consists of the BIGG ontology serving as a common reference, a Harmonizer tool enabling automated data harmonization of heterogeneous data sources over the BIGG ontology, and includes mapping and transformations for the different data sources used in the project.

The process of development of the harmonization layer consisted of several steps. The first step was the development of the BIGG Standard Data Model 4 Buildings based on detailed analysis of requirements of the BIGG use cases, and of the available datasets from the pilots necessary for the execution. The data model set the semantic base and structure of data in BIGG and served as a common reference for the parallel work in the work packages of the project dealing with communication, data analytics, and integration of components in the reference architecture.

In a next step, the Standard Data Model 4 Buildings step was transformed into a W3C standards compliant BIGG Ontology based on the RDF specifications, thus enabling the use of semantic technologies and machine understanding of data.

The adoption of the RDF as internal format for representing data in BIGG enabled the development of the Harmonizer component, a generic tool for converting both static and dynamic building-related data into BIGG-compliant data in RDF. The Harmonizer allows to use RML mapping rules to align the input data to the BIGG ontology and also implements SPARQL queries to specify correspondence between standard ontologies (e.g., ifcOWL) and the BIGG ontology.

Finally, aiming to contribute to standardisation, the BIGG Ontology was substantially analysed and compared to existing standards. This led to the development of BIGGstd, a standards-based transformation of the BIGG Ontology obtained by reusing existing standards. The transformation process served to evidence gaps in existing ontologies and to pinpoint potential contributions to them. A set of specific terms defined by BIGG for which no equivalent terms in existing standards were found was identified and presented in Annex 1. The potential use of these terms for future extensions of existing data standards will be considered by the BIGG representatives in EU standardization committees.

In addition to the overview of the harmonization layer components above, the document provides links to their full documentation uploaded in the public GitHub repository of the project.

Table of Contents

Table of Figures

Table of Acronyms and Definitions

Acronym	Definition
AI	Artificial Intelligence
EC	European Commission
IoT	Internet of Things
JSON	JavaScript Object Notation
OWL	Web Ontology Language
RDF	Resource Description Framework
RML	RDF Mapping Language
TTL	Terse RDF Triple Language
UML	Unified Modelling Language
W3C	World Wide Web Consortium
WP	Work Package

Contributors Table

<table>
<tr><th>DOCUMENT SECTION</th><th>AUTHOR(S)</th><th>CONTRIBUTOR(S) to results</th><th>REVIEWER(S)</th></tr>
<tr><td>All sections</td><td>Stoyan Danov (CIMNE)</td><td>Eloi Gabaldon (CIMNE), Jordi Carbonell (CIMNE), Edgar Martínez (CIMNE),
Guillaume Picinbono (CSTB),
Nicolas Bus (CSTB)</td><td rowspan="3">Chris Develder (imec)</td></tr>
<tr><td>Section III</td><td>Stoyan Danov (CIMNE)</td><td>Eloi Gabaldon (CIMNE), Jordi Carbonell (CIMNE), Edgar Martínez (CIMNE)</td></tr>
<tr><td>Section IV</td><td>Guillaume Picinbono (CSTB)</td><td>Nicolas Bus (CSTB)</td></tr>
</table>

I. INTRODUCTION

The aim of this report is to present the final harmonization layer development and results.

The harmonization layer in BIGG is essential part of the BIGG reference architecture that enables semantic exchange of data with external data hubs, defines a common data model for internal harmonized data storage within the BIGG applications, and ensures unified data input to the AI toolbox.

The BIGG harmonization layer was developed in two stages. The first stage took place during the first year of the project and produced as a result the BIGG Standard Data Model 4 Buildings. The initial development was based over detailed analysis of the requirements of the BIGG use cases and conceptualization and the model description was formalized through UML diagrams and a detailed description of classes, attributes, and relationships. The methodological approach for development and the BIGG Standard Data Model 4 Buildings itself are reported in deliverable D4.1. This work was key for ensuring the alignment between the analytics, storage, and communication components of the BIGG architecture and for the implementation of the first version of demonstrators. At this stage, the harmonization of data from external sources involved manual mapping to the BIGG Standard Data Model 4 Buildings and customized transformations.

The second stage of the harmonization layer development covered by this document aims to automate the data harmonization process and to advance into the alignment with standards. At this stage the Standard Data Model 4 Buildings was transformed into BIGG ontology following the World Wide Web Consortium (W3C) standards of the Semantic Web based on the Resource Description Framework (RDF) specifications. An overview of the BIGG ontology is presented in Section III.

The adoption of sematic technologies and RDF as the internal data format for the BIGG architecture enabled the development of the BIGG Harmonizer. This component automates the conversion of external data formats into the harmonized RDF format following the BIGG ontology. The BIGG Harmonizer is presented in Section IV.

During this stage, considerable effort was put into reusing existing standards in the BIGG ontology, which led to the development of the BIGG standards-based ontology (BIGGstd). This work permitted the identification of gaps in current standards and highlighted potential extensions for them. An overview of the BIGGstd ontology is presented in section V.

The full definitions of the BIGG ontology, the Harmonizer, and the BIGGstd ontology are published in the public GitHub repository of the project: https://github.com/biggproject.

The specific concepts (classes, data properties, and object properties) in the BIGGstd ontology for which no equivalent terms were identified in existing ontologies are presented in Annex I. These particular concepts represent the potential contribution of BIGG to the future extension of existing standards that will enable them to support the BIGG use cases and tools.

I.1. Organization of the document

This report is organized as follows:

- Section II describes the approach for data harmonization in BIGG and the steps of its implementation.
- Section III presents the BIGG Ontology supporting the data harmonization and interoperability between the components of the BIGG reference architecture.
- Section IV presents the BIGG Harmonizer and its functionalities.
- Section V presents the work on transforming the BIGG Ontology into standards-based ontology BIGGstd and the contribution to existing standards.
- Section VI provides conclusions and outlines future activities.
- Annex I provides the description of the new concepts introduced by BIGG that could be considered for extending the existing standards.

I.2. Scope and audience

This document is important for all project participants and users of the BIGG tools. It provides an overview of, and references to, the developed BIGG ontology which constitutes the base for alignment and exchange within the BIGG reference architecture. The report is especially relevant for implementers and providers of data-driven services related to buildings that are interested in the use of the BIGG AI Toolbox. It also provides interesting information for researchers and practitioners engaged in developing ontologies and data standards. The presented work in reusing and aligning to existing ontologies in the context of the wide scope of applications covered by BIGG has permitted to highlight gaps in existing standards and has formulated proposals of new concepts to be considered for inclusion in future standards.

II. APPROACH TO DATA HARMONIZATION

II.1. Context

Buildings represent the largest share of European final energy consumption (40%), according to the "Energy Efficiency – first fuel for the EU economy" report **Error! Reference source not found.**. Buildings also present the greatest potential to save energy, as 75% of those standing in the EU were built during periods with minimal energy-related regulations and 75-90% of those standing today are expected to remain in use in 2050. Even in new construction and energy retrofits, a significant gap between the expected and the actual, measured consumption in operation is observed **Error! Reference source not found.**. Human behavioural factors, such as occupant behaviour, and the quality of provided indoor environmental conditions are pointed out as factors affecting the energy consumption to an extent at least as great as those of climate, building envelope, and energy systems characteristics **Error! Reference source not found.**.

Big data technologies, together with the development of the Internet of Things (IoT), opened a whole scope of new possibilities. The continuously increasing amount of data generated by buildings through the adoption of digital technologies, integrating sensors, controllers, and their connectivity, offer enormous potential for increasing the efficiency in both existing and new buildings. However, this potential is hindered by the ambiguity of the data definitions and lack of standardization across applications and databases, which makes it difficult to exchange, compare, and combine the data, both at building and inter-building levels. As a result, (1) only a small fraction of the available data is analysed and effectively used for providing innovative building-related services; (2) available data is often used in silos and cannot be combined with other data or employed in other use cases.

The harmonization of data by BIGG aims to contribute to overcoming these obstacles. In this context, the BIGG project aims to facilitate the implementation of big data analytics for buildings, reducing the effort to create applications. The project focuses on the development of an open-source Big Data Reference Architecture and AI Analytics Toolbox with demonstrated capabilities to support a variety of use cases and applications covering the whole building life cycle. Combining of data from different sources for joint analytics and standardized input and output from the Analytics Toolbox components are indispensable features for the solution. Therefore, data harmonization and interoperability are core to the concept of BIGG.

Whereas technical interoperability, i.e., the successful exchange of data, can be more easily achieved through the adoption of standardized communication protocols, the lack of harmonization at the information level, which deals with the semantic (meaning) of data, constitutes a more complicated issue [4]. Semantic technologies, a combination of software and specifications that allow encoding the meaning of data and their interrelations in a machine-processable form, offer powerful tools to overcome the interoperability challenge [5]. Semantic models exist in different levels of complexity, including ontologies or knowledge graphs, and data models that represent sets of concepts belonging to a specific domain and the relationships between them [6]. Previous research has shown that the association of raw data to terms belonging to a common ontology, and thus a common meaning, facilitates the uniform representation of data collected by different sources and therefore their informational interoperability **Error! Reference source not found.**, **Error! Reference source not found.**.

Most of the currently developed ontologies follow the World Wide Web Consortium (W3C) standards for the Semantic Web and are aimed at making internet data machine-readable. W3C standards-compliant ontologies are based on the Resource Description Framework (RDF) specifications, according to which information can be modelled combining triples in the form of "subject – predicate – object", expressions called "triples" in RDF terminology **Error! Reference source not found.**. Built on top of RDF, the Web Ontology Language (OWL) is designed for defining and instantiating Web ontologies.

II.2. The BIGG approach

Data harmonization is a process of bringing together data of varying file formats, naming conventions, and columns, and transforming them into one cohesive dataset. In the context of the BIGG Reference Architecture Framework, the harmonization layer ensures the capability to align, harmonize, and make comparable data from different sources over an internal standard data model. Such harmonized data format is adopted for the AI Analytics Toolbox components' input, enabling the integration of the developed tools with legacy systems and technologies, as well as with new developments.

The harmonization layer aims to make the process of data harmonization systematic, reproducible, and operational. It is materialized through the development of the BIGG Standard Data Model 4 Buildings, open documentation, and specifications for implementation of open-source components for data mapping and harmonization.

The first step was the development of the BIGG Standard Data Model 4 Buildings. Its development enabled the coordinated implementation of the tasks in the different work packages, facilitating the integration and use of the AI Toolbox in the demonstrations.

The second step was the evolving of the data model into a W3C-standards-compliant BIGG ontology based on the RDF specifications, thus making possible the use of semantic technologies, enabling advanced representation and machine understanding of data. At this stage, the naming conventions of the BIGG Standard Data Model 4 Buildings were followed, keeping them as a reference to facilitate the parallel work in developing the different components of the BIGG architecture. During this step, the BIGG Harmonizer component for automated harmonization of data was also developed. This component orchestrates the transformation of heterogeneous external data to the harmonized RDF format and is extensible to any data provider.

As a final step, with the objective to contribute to standardisation, the BIGG ontology was transformed into standards-based ontology by reusing existing standards. The process was iterative and comprised detailed analysis of existing ontologies, selection of equivalent terms, and reasoning on their adoption in BIGG. The process resulted in the creation of the BIGGstd ontology and served to evidence gaps in existing ontologies and to pinpoint potential contributions to them.

The harmonization layer development stages are illustrated in Figure 1 below. The development of the BIGG Standard Data Model 4 Buildings (version 1) corresponds to the initial phase and is presented in deliverable "D4.1 - Description of the preliminary harmonization layer". The second and third steps (version 2) have been developed in the second phase and presented in the current document.

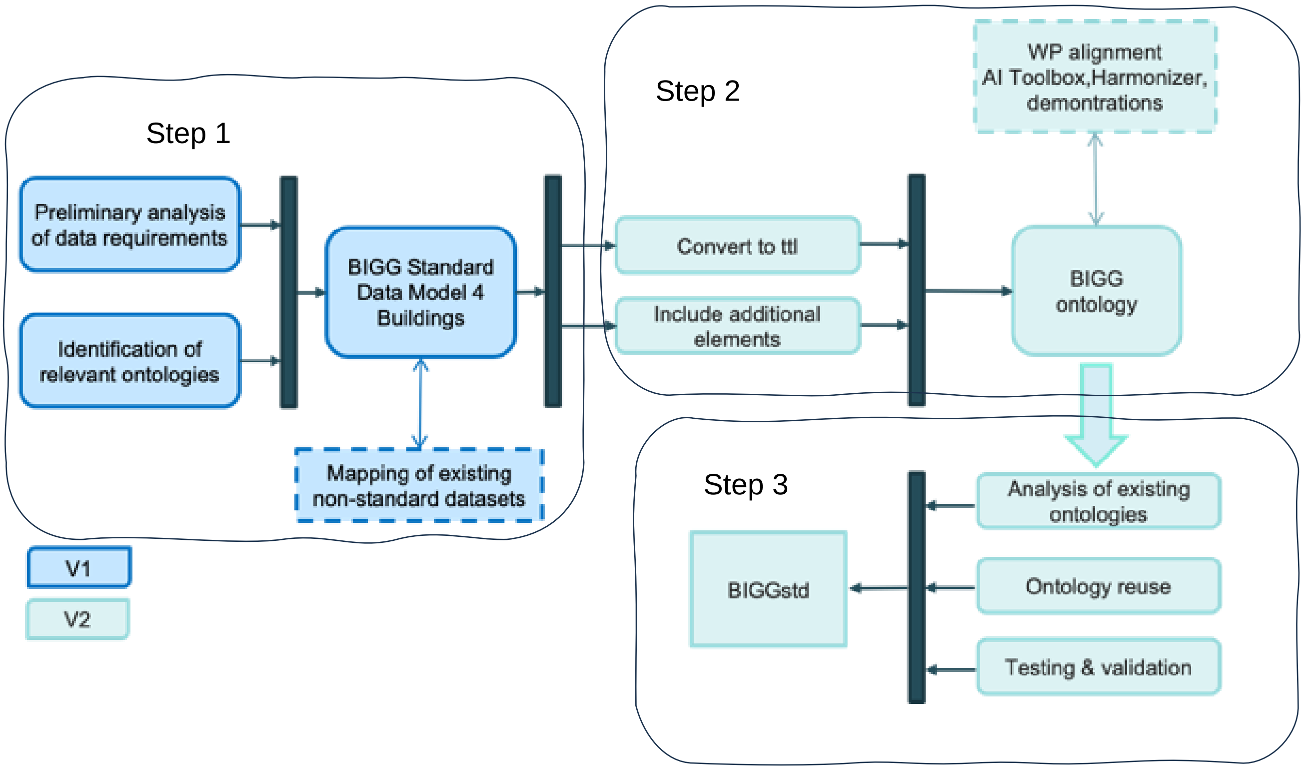


Figure 1 - Steps in the development of the BIGG harmonization layer

III. THE BIGG ONTOLOGY

This section presents the operative version of the BIGG ontology that is used for the implementation of the use case demonstrations with support the BIGG AI Toolbox. The ontology was created over the conceptual base of the BIGG Standard Data Model 4 Buildings and served as reference for the integration and communication between the components in the BIGG architecture. To facilitate parallel work in the BIGG Work Packages, the ontology strictly followed the naming conventions of the BIGG Standard Data Model 4 Buildings, and only minor changes and additions were made.

The BIGG ontology was created in Terse RDF Triple Language (TTL), a W3C standard format, over the conceptual base and naming conventions of the BIGG Standard Data Model 4 Buildings.

The following subsections provide a high-level overview of the ontology structure, main classes, and relations between them. The full documentation of the ontology is available in the public GitHub repository of the BIGG project (https://github.com/biggproject).

III.1. General overview

The BIGG ontology has a modular structure consisting of a core and extensions. The core includes three essential class groups, and the extensions provide additional classes and relations. This structure is illustrated in Figure 2.

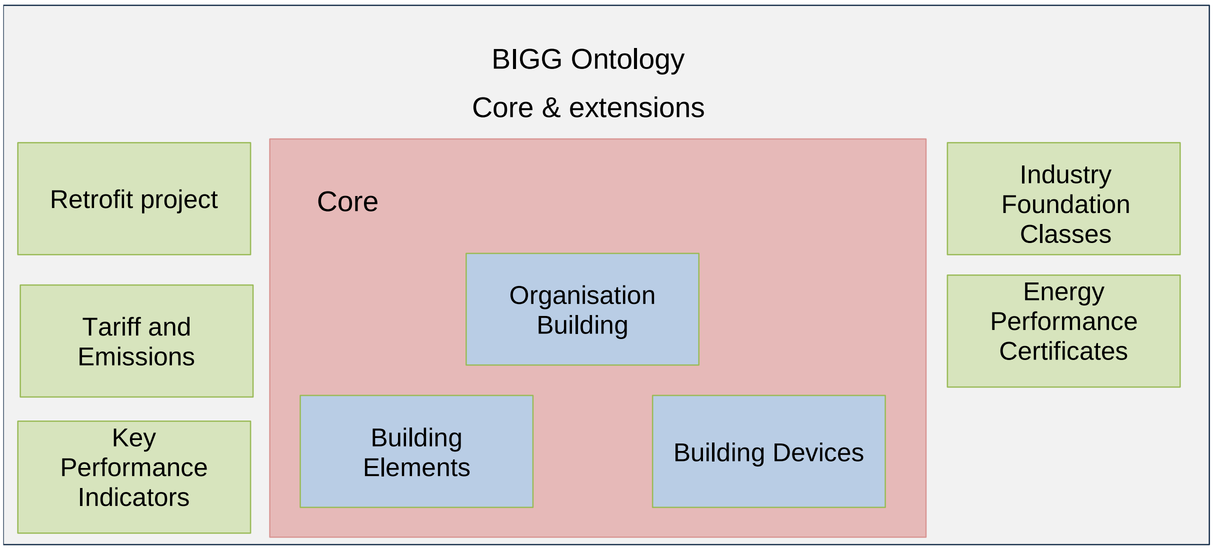


Figure 2 - Structure of the BIGG ontology

III.1.1. Core

The core of the BIGG ontology consists of three modules named over the main classes contained in them. These modules are presented below in a structured way through class diagrams, classes and attributes, subclass definitions, and namespaces used in the ontology.

Module: Organization - Building

This module defines the relations between the classes bigg:Building and bigg:Organization and other related to them classes.

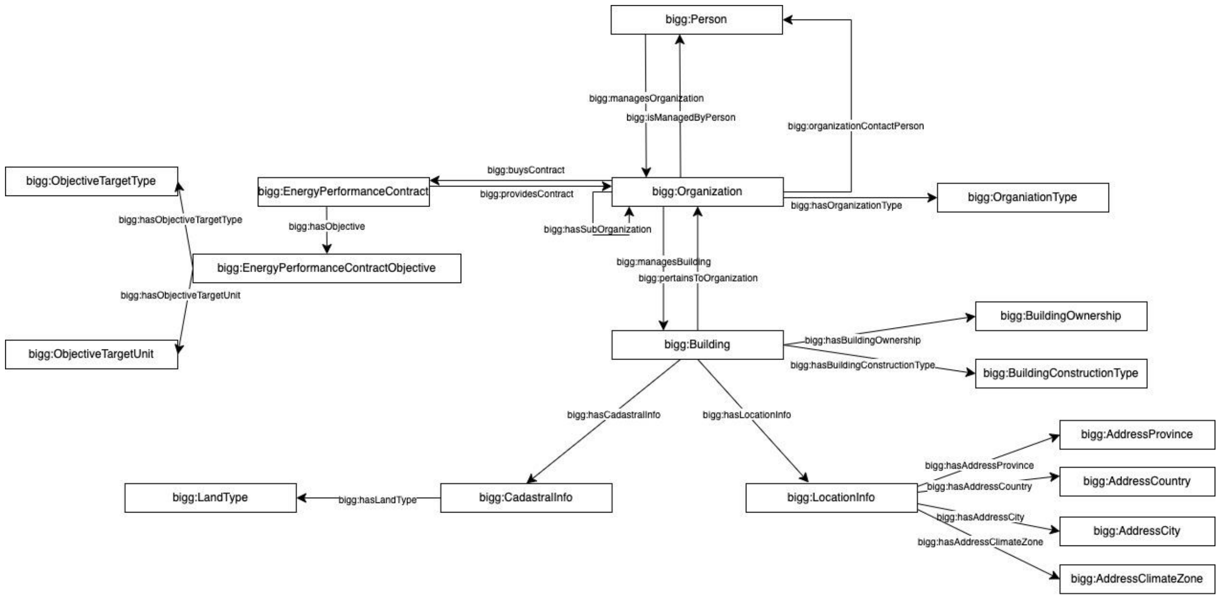
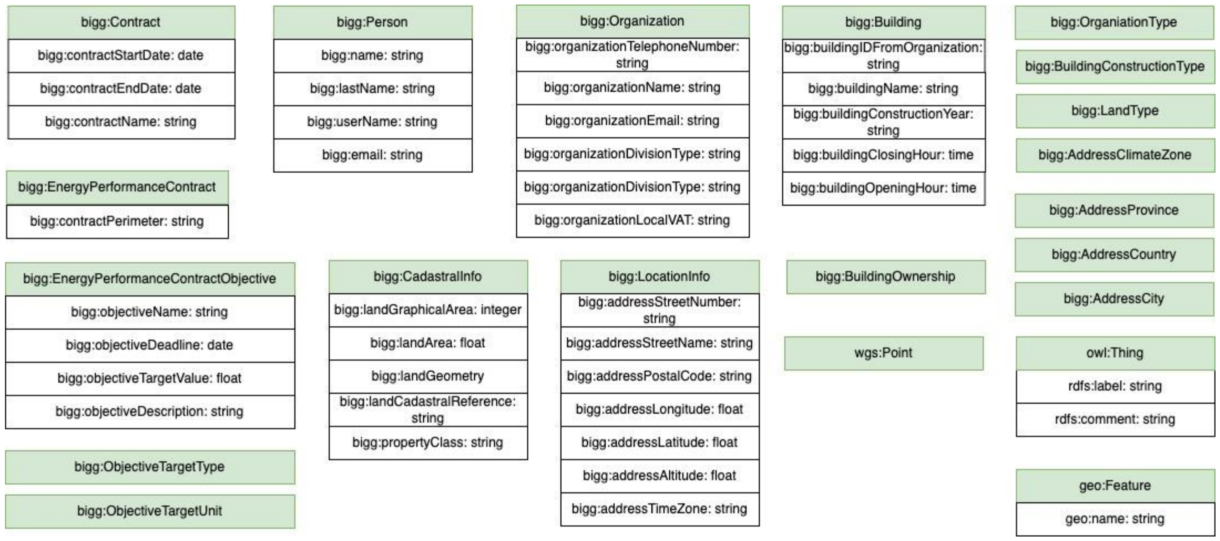
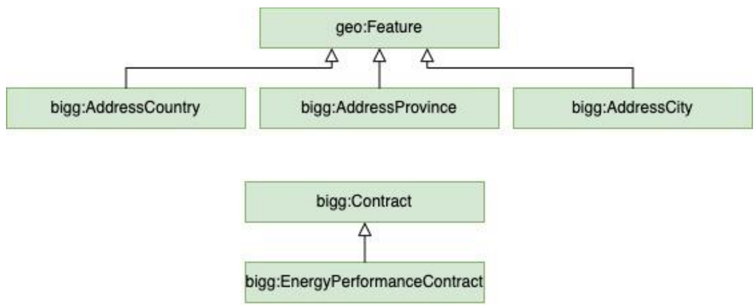


Figure 3 - Module: Organization - Building

Classes and data properties:



The following diagrams provide additional information about classes, subclasses, and namespaces used in the ontology.



Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
wgs: http://www.w3.org/2003/01/geo/wgs84_pos#

Module: Building - Element

This module defines the relations between bigg:Building, bigg:Element and the classes connected to them.

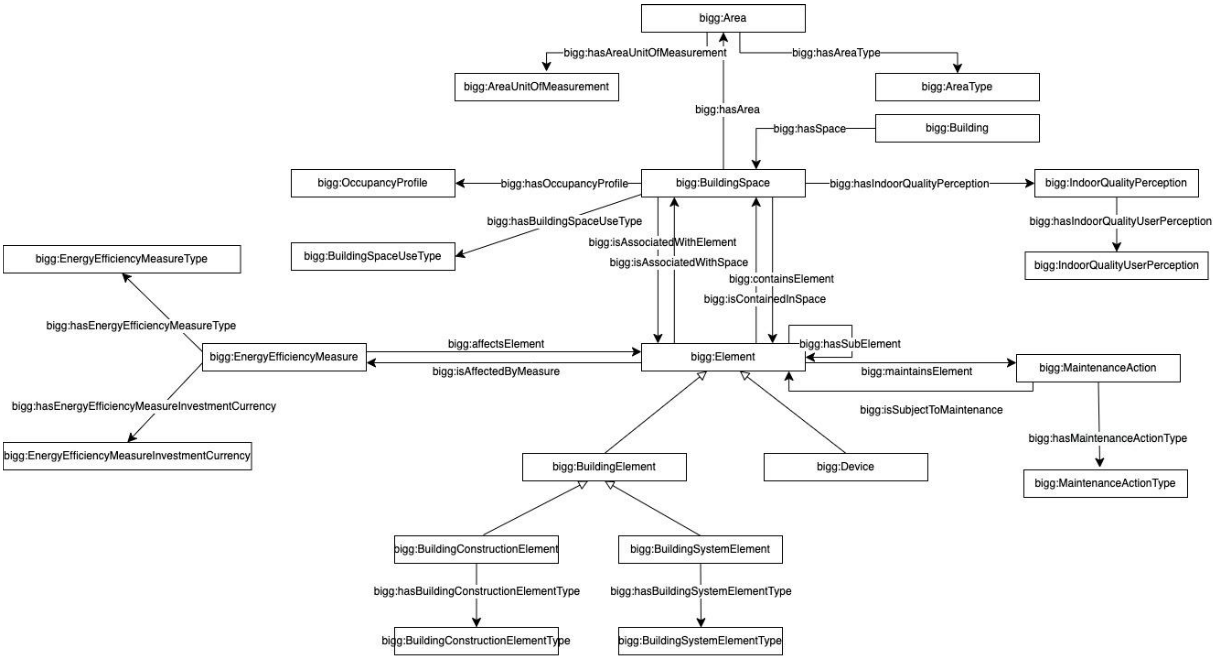
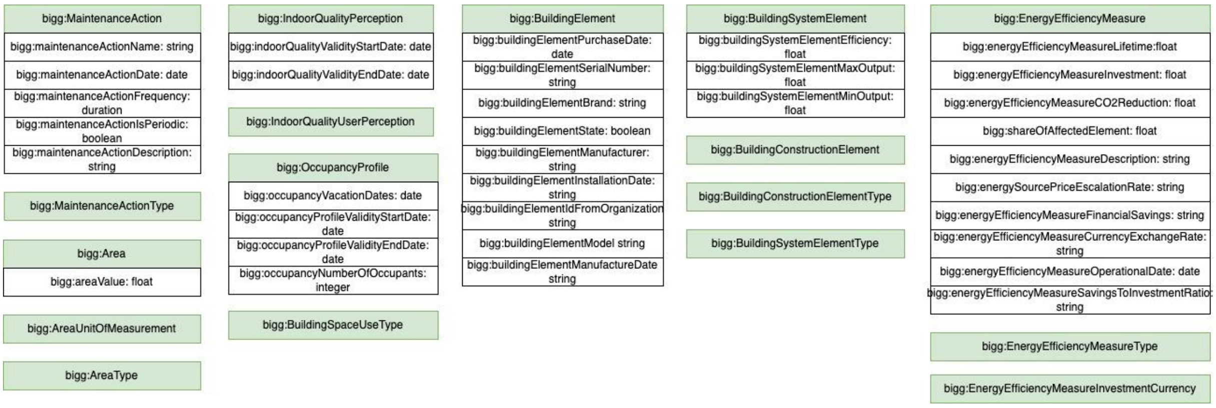


Figure 4 - Module: Building - Element

Classes and data properties:



Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
wgs: http://www.w3.org/2003/01/geo/wgs84_pos#

Module: Building - Device

This module defines the relations between the classes bigg:Building and bigg:Device. The classes related to them, as needed for the BIGG use cases, are also defined.

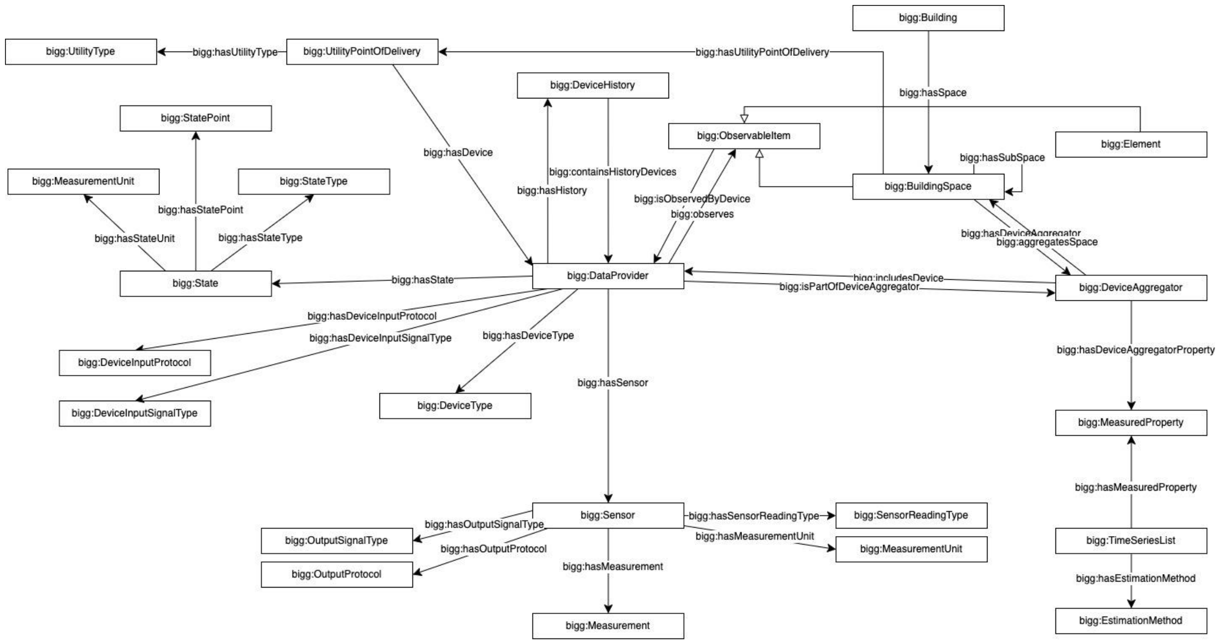
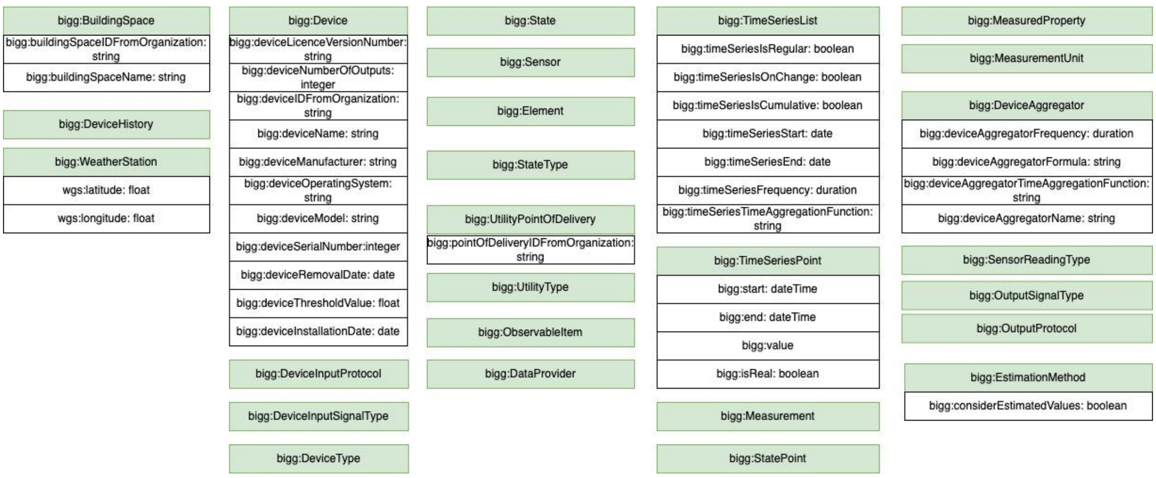
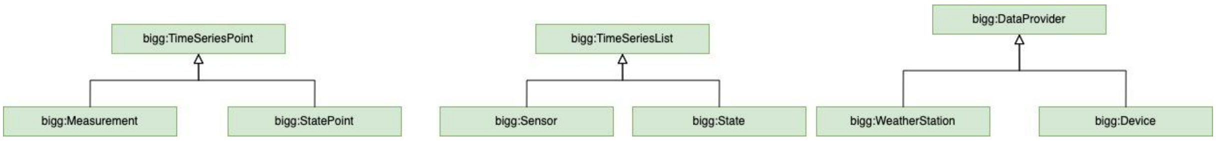


Figure 5 - Module: Building - Device

Classes and data properties:



Subclass definitions:



Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
wgs: http://www.w3.org/2003/01/geo/wgs84_pos#

III.1.2. Extensions

Module: Retrofit project

This module defines the extension for retrofit projects and relations of the main concept bigg:RetrofitProject with other concepts in the context of energy efficiency retrofitting.

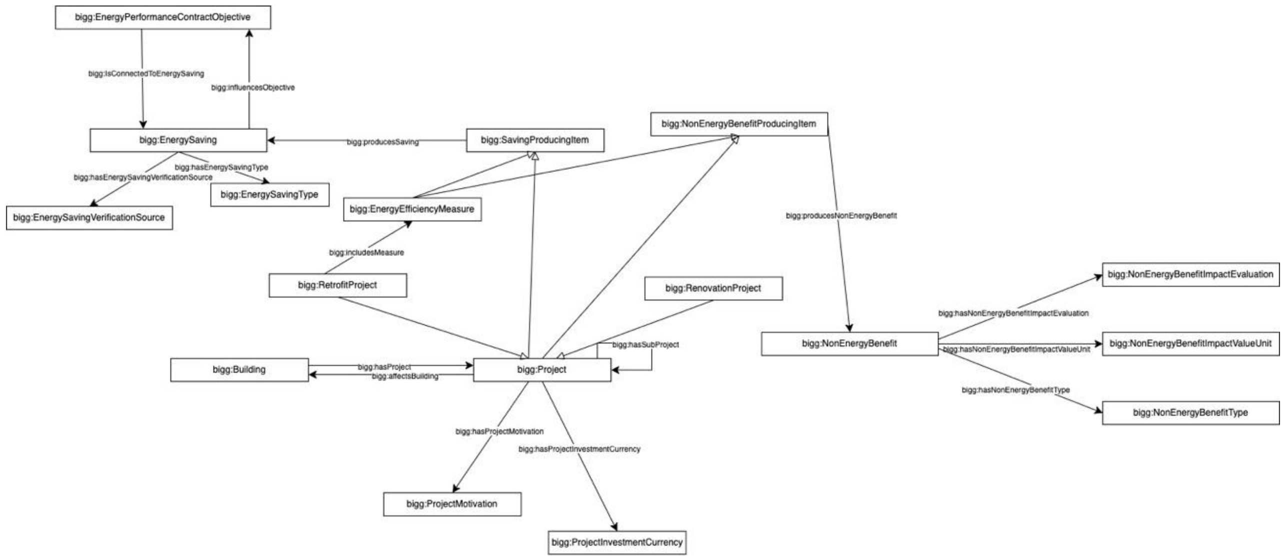


Figure 6 - Module: Retrofit project

Classes and data properties:

bigg:Project
bigg:projectIDFromOrganization: string
bigg:projectName: string
bigg:projectDescription: string
bigg:geometrySRID: string
bigg:projectDiscountRate: float
bigg:projectSimplePaybackTime: float
bigg:projectOperationalDate: date
bigg:projectStartDate: date
bigg:projectIncludedConfortmeterSurvey: boolean
bigg:projectInterestRate: float
bigg:projectGrantsShareOfCosts: float
bigg:projectInternalRateOfReturn: float
bigg:projectIncludedNonEnergyBenefitsEstimate: boolean
bigg:projectCurrencyExchangeRate: float
bigg:projectUsesIncentives: boolean
bigg:projectNetPresentValue: float
bigg:projectInventivesShareOfRevenues: float
bigg:projectReceivedGrantFounding: boolean
bigg:projectSavingsToInvestmentRatio: float
bigg:projectInvestment: float

bigg:RenovationProject

bigg:RetrofitProject
bigg:projectCO2Reduction: float

bigg:SavingProducingItem

bigg:NonEnergyBenefitProducingItem

bigg:ProjectInvestmentCurrency

bigg:ProjectMotivation

bigg:NonEnergyBenefit
bigg:nonEnergyBenefitImpactVerificationMethod: string
bigg:nonEnergyBenefitImpactValue
bigg:nonEnergyBenefitImpactValueVerifiedAndMeasured: boolean
bigg:nonEnergyBenefitImpactValueDescription: string

bigg:NonEnergyBenefitImpactEvaluation

bigg:NonEnergyBenefitImpactValueUnit

bigg:NonEnergyBenefitType

bigg:EnergySaving
bigg:energySavingStartDate: string
bigg:energySavingEndDate: string
bigg:energySavingValue: float
bigg:energySavingIndependentlyVerified: boolean

bigg:EnergySavingVerificationSource

bigg:EnergySavingType

Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
geo: http://www.geonames.org/ontology#

Module: Tariff and emissions

This module describes the extension for energy tariff and emissions and the relations around the concepts bigg:Tariff and bigg:CO2EmissionaFactor.

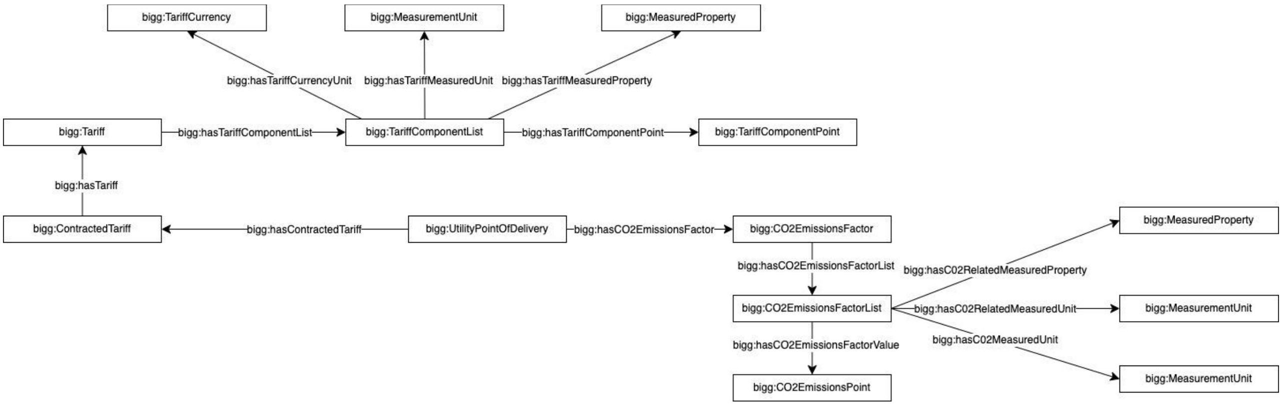
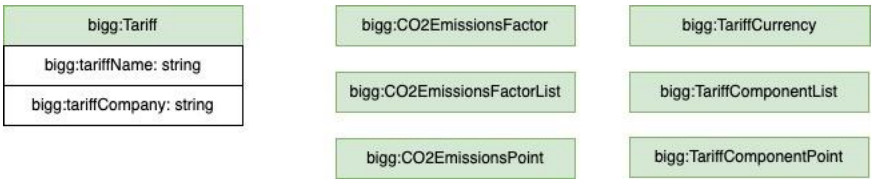
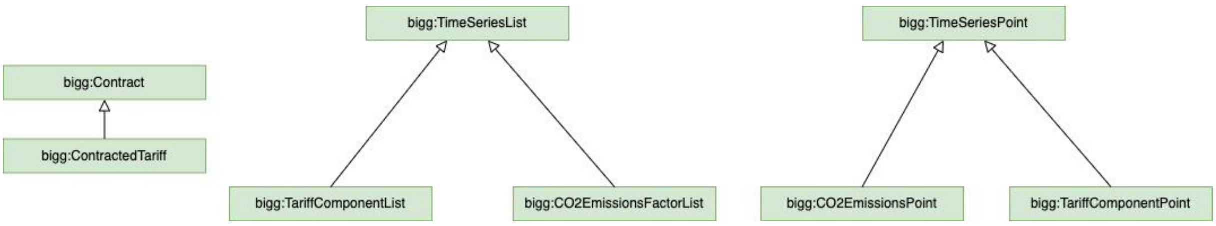


Figure 7 - Module: Tariff and emissions

Classes and data properties:



Subclass definitions:



Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
geo: http://www.geonames.org/ontology#

Module: Key Performance Indicators (KPI)

This describes the extension for KPIs and their relations.

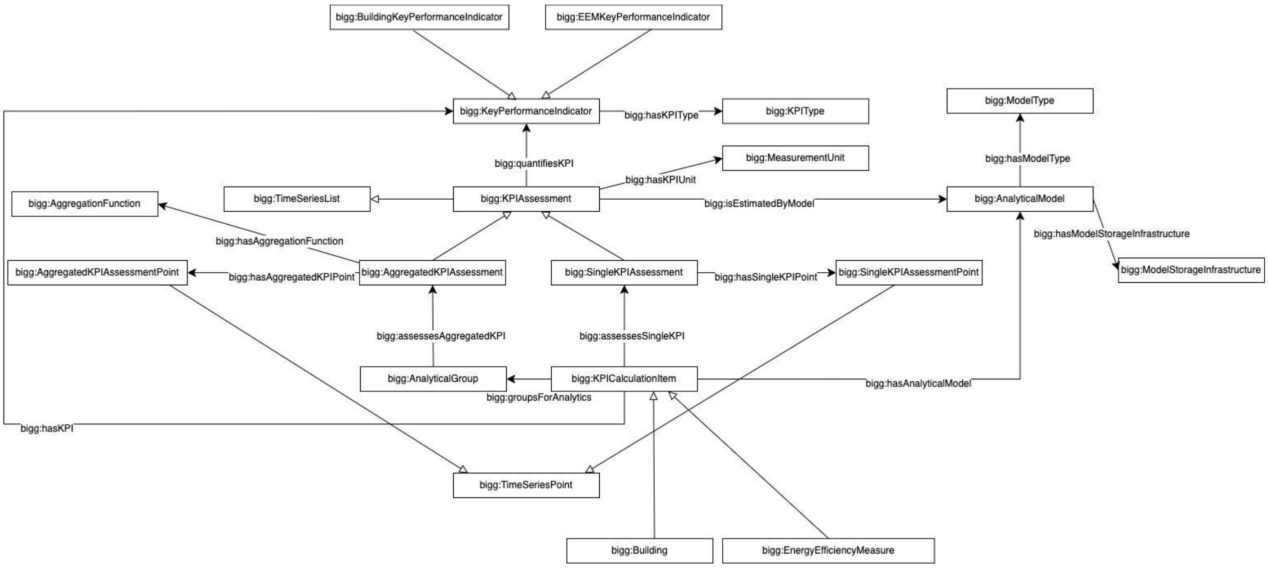


Figure 8 - Module: Key Performance Indicators

Classes and data properties:

- bigg:KPICalculationItem
- bigg:KeyPerformanceIndicator
- bigg:BuildingKeyPerformanceIndicator
- bigg:EEMKeyPerformanceIndicator
- bigg:KPIType
- bigg:KPIAssessment
- bigg:AggregatedKPIAssessment
- bigg:AggregatedKPIAssessmentPoint
- bigg:SingleKPIAssessmentPoint
- bigg:AnalyticalGroup
- bigg:ModelStorageInfrastructure
- bigg:SingleKPIAssessment

bigg:AnalyticalModel
bigg:modelName: string
bigg:modelLocation: string
bigg:modelTrainedDate: date
bigg:modelBaselineYear: string

Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
geo: http://www.geonames.org/ontology#

Module: Industry Foundation Classes (IFC)

This module describes the extension with concepts and relations adopted from the IFC schema.

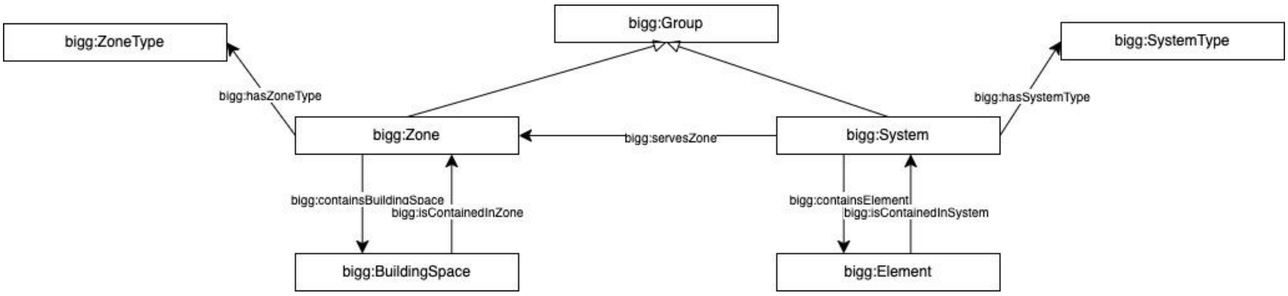


Figure 9 - Module: Industry Foundation Classes

Classes and data properties:

bigg:Zone

bigg:System

bigg:ZoneType

bigg:SystemType

bigg:Group

bigg:groupName: string

Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
geo: http://www.geonames.org/ontology#

Energy Performance Certificate (EPC)

This extension defines the concepts and relations for Energy Performance Certificates.

Figure 10 - Module: Energy Performance Certificates

Classes and data properties:

bigg:EnergyPerformanceCertificate
bigg:energyPerformanceCertificateReferenceNumber: string
bigg:energyPerformanceCertificateProcedureType: string
bigg:energyPerformanceCertificateCertificationTool: string
bigg:energyPerformanceCertificateClass: string
bigg:energyPerformanceCertificateCertificationMotivation: string
bigg:energyPerformanceCertificateDateOfCertification: date
bigg:energyPerformanceCertificateDateOfAssessment: date
bigg:C02EmissionsClass: string
bigg:annualC02Emissions: float
bigg:annualEnergyCost: float
bigg:annualPrimaryEnergyConsumption: float
bigg:annualFinalEnergyConsumption: float
bigg:lightingCO2EmissionsClass: string
bigg:annualLightingCO2Emissions: float
bigg:lightingPrimaryEnergyClass: string
bigg:lightingPrimaryEnergyConsumption: float
bigg:heatingCO2EmissionsClass: string
bigg:annualHeatingCO2Emissions: float
bigg:heatingPrimaryEnergyClass: string
bigg:annualHeatingPrimaryEnergyConsumption: float
bigg:annualHeatingEnergyDemand: float
bigg:heatingEnergyDemandClass: string
bigg:hotWaterCO2EmissionsClass: string
bigg:annualHotWaterCO2Emissions: float
bigg:hotWaterPrimaryEnergyClass: string
bigg:annualHotWaterPrimaryEnergyConsumption: float
bigg:coolingCO2EmissionsClass: string
bigg:annualCoolingCO2Emissions: float
bigg:coolingPrimaryEnergyClass: string
bigg:annualCoolingPrimaryEnergyConsumption: float
bigg:annualCoolingEnergyDemand: float
bigg:coolingEnergyDemandClass: string

bigg:EnergyPerformanceCertificateAdditionalInfo
bigg:buildingTechnicalInspectionCode: string
bigg:constructionRegulation: string
bigg:electricVehicleChargerPresence: boolean
bigg:biomassSystemPresence: boolean
bigg:geothermalSystemPresence: boolean
bigg:districtHeatingOrCoolingConnection: boolean
bigg:solarThermalSystemPresence: boolean
bigg:solarPVSystemPresence: boolean
bigg:averageFacadeTransmittance: float
bigg:averageWindowsTransmittance: float
bigg:regulationValueForWindowsTransmittance: float
bigg:regulationValueForFacadeTransmittance: float

Namespaces:

bigg:http://bigg-project.eu/ontology#
rdfs: http://www.w3.org/2000/01/rdf-schema#
skos: http://www.w3.org/2004/02/skos/core#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd: http://www.w3.org/2001/XMLSchema#
geo: http://www.geonames.org/ontology#

III.2. Public documentation

The BIGG ontology and its full documentation is available on the public GitHub repository of the BIGG project: https://github.com/biggproject/Ontology/tree/main/BIGG.

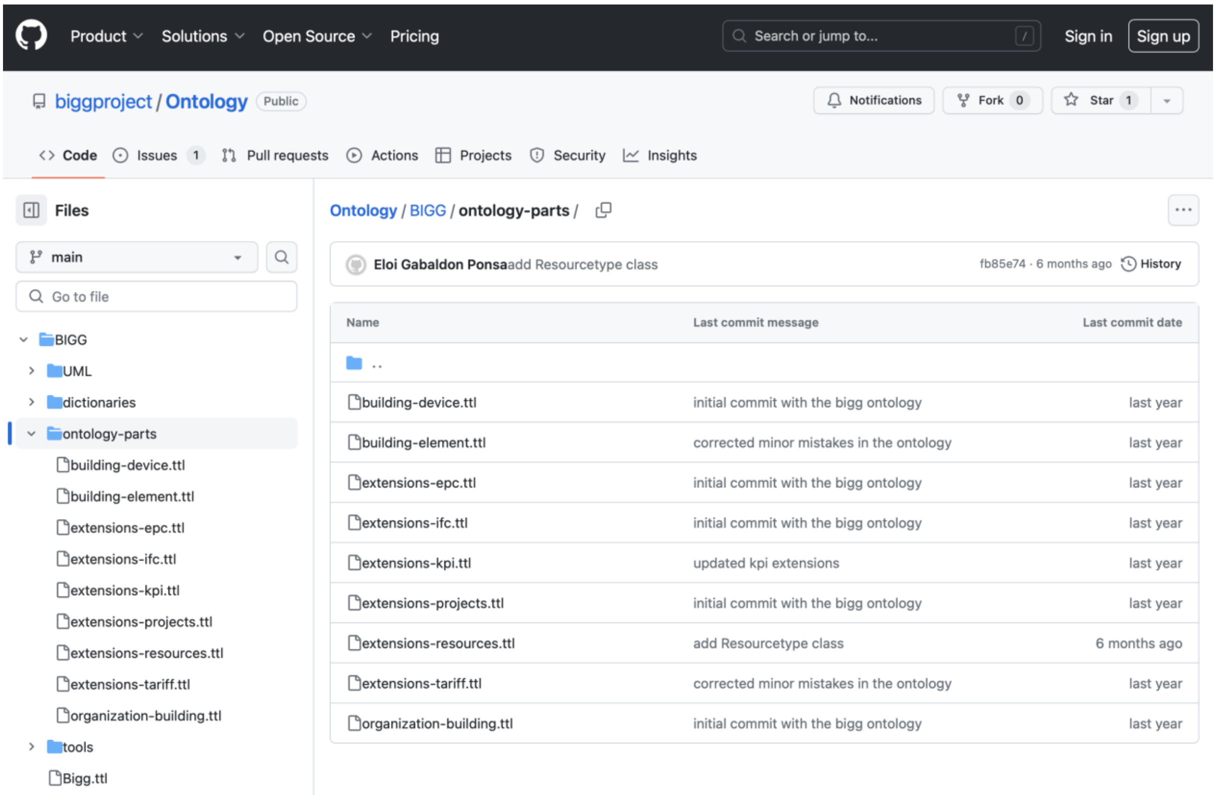


Figure 11 - BIGG ontology documentation on GitHub

IV. BIGG HARMONIZER

Data harmonization serves as a foundational step toward realizing data-driven excellence, ensuring that data evolves from a potential 'obstacle' into an 'asset' that propels an organization's operations and growth endeavours.

The BIGG use cases involve a multitude of data models and formats, coming from different sources such as open data providers, assets management software and sensors. Regarding data formats, relational databases and XML are still present, Open Data portals heavily rely on CSV, and web APIs on JavaScript Object Notation (JSON).

In BIGG, the RDF data model is adopted for internal harmonized representation of the data. The mapping of the external data having heterogeneous formats to the BIGG semantics and their conversion into the harmonized RDF format of BIGG allows the data interoperability and enables the use of the BIGG AI Toolbox. To simplify the mapping specification for the data providers and avoid the effort to code serializers for each kind of source using a generic programming language, the harmonizer is based on RML (RDF Mapping Language).

IV.1. Implementation of the harmonizer tool

The BIGG Harmonizer is a generic tool implemented in Python that serves to convert any building's energy-related data into data compliant with a BIGG digital twin. It is created to harmonize input data with the BIGG ontology. The process of harmonization comprises two stages: the first one is related to the RML mapping file processing, and the second one is related to the alignment of standards ontologies with the BIGG ontology.

A schematic of the tool is presented in Figure 12 below.

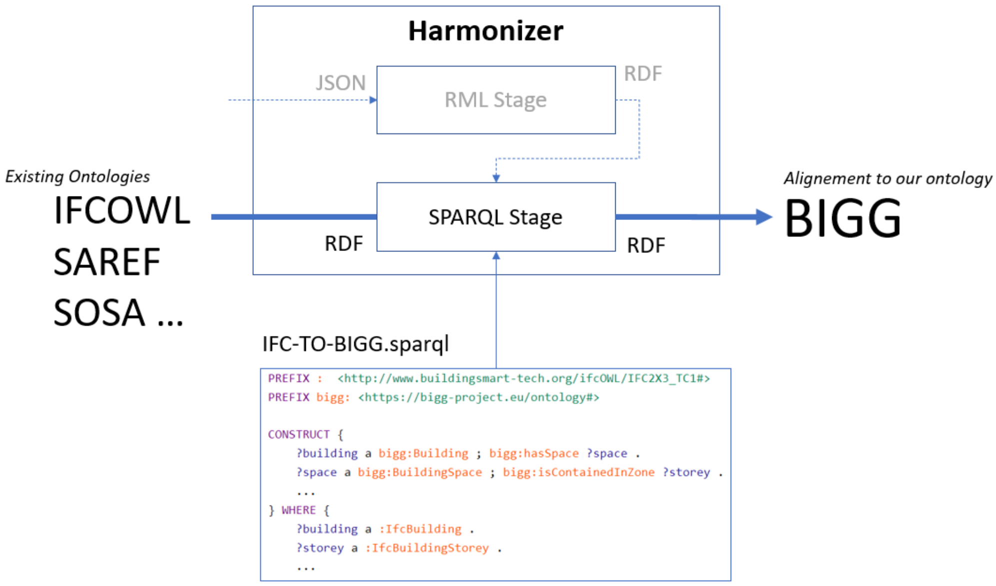


Figure 12 - BIGG Harmonizer tool

The conversion step is the first stage of the Python module. It corresponds to the use of the java library to convert an input JSON file into an RDF file, thanks to the mapping rules defined in the RML file. The module can output two serialization formats of RDF: the TTL (Turtle) or the JSON-LD format.

The second stage of the Python module corresponds to the use of SPARQL queries to add, translate or complete the RML stage. The second stage can also be used to align data based on standardized ontology into BIGG-compliant RDF. For instance, it allows to align data based on standards ontologies such as ifcOWL, SOSA or SAREF, with the BIGG ontology.

The execution and test of the Python module can be done in a Jupiter Notebook, with the following command line:

```
python harmonizer.py --input inputFile [--mapping RMLFile] [--sparql SparqlFiles]
[--output outputFilename]
```

The harmonizer process can be illustrated with a simple example of a building containing storeys, spaces, and some devices. Figure 14 shows the corresponding JSON input file on the top left, and the YARRRML [13] mapping file on the right. Once the RML mapping file is generated, we can use it to convert the input file into an RDF file aligning with the BIGG Ontology (bottom left).

The Python module allows the use of the RML mapping rules to align the input data with the BIGG ontology into an RDF document. Automatically, it generates instances of BIGG objects like bigg:Building, bigg:BuildingSpace and bigg:Device with the related relations.

The same approach has been successfully applied to input data from other sources: group, systems, and sensors structures exported from exploitation control software, or time series coming directly from sensors. Each specific data is mapped to instantiate a specific part of the BIGG ontology.

In a second phase, the harmonization module is also used to create rules to include import data in standard format (e.g., ifcOWL) into an instance of a BIGG model. Indeed, the harmonizer's second stage allows to execute SPARQL queries to specify correspondences between ontologies.

The diagram in Figure 13 provides a detailed sequence of the processing logic and steps.

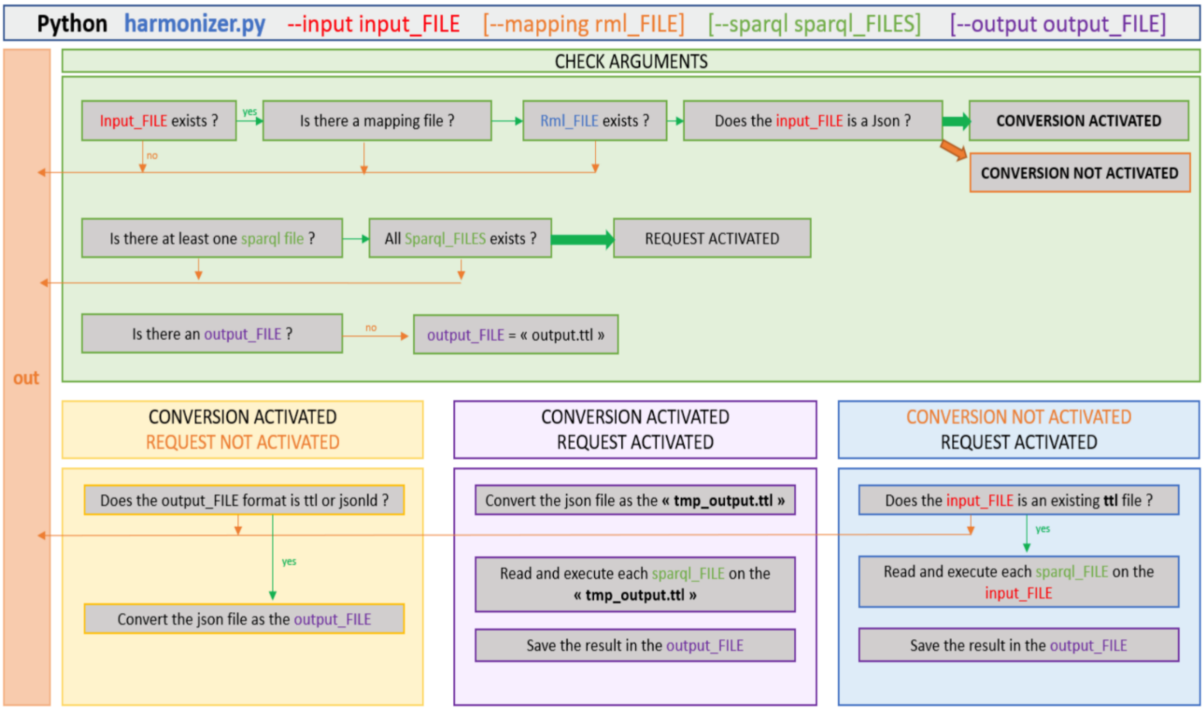


Figure 13 - Harmonizer processing flow

IV.2. Development of mappings

This section describes the choice of the main mapping language in BIGG and the different implementations of mapping realized by CSTB and CIMNE within the demonstrations.

IV.2.1. Choice of mapping language

Several solutions have been proposed to map semi-structured data to RDF. The SweoIG taskforce has built a list of implementations also called RDFizer [8]. Most of them are dedicated to a specific format or model. We focused our analysis on those that fulfill the project requirements: a language that can be extended with calls to user-defined functions, that is model agnostic, and fully compliant with W3C recommendations.

RML [9] is one of the approaches that allows mapping between various sources toward RDF. RML is an extension of the R2RML vocabulary to describe logical sources which are different from relational database tables. It generates RDF from JSON by using JSONPath but can also handle CSV or XML sources.

SPARQL-GENERATE [10] is based on a query language that queries the combination of an RDF dataset and a set of documents. Various formats can be supported thanks to the extensible set of SPARQL 1.1 binging functions and SPARQL-Generate iterator functions. SPARQL-Generate is appropriate for engineers that are familiar with SPARQL as it built one top of SPARQL [11].

OpenRefine [12] (formerly Google Refine) helps to make sense of tabular data through a set of tools to clean, transform, and cross-link several datasets together. RDFRefine is an extension of OpenRefine that is meant to align the tabular data loaded in OpenRefine with existing ontologies, while reconciliating the obtained URIs with third party RDF datasets. In this context, RDF Refine provides an export feature that translates tabular data into RDF. The data management part of OpenRefine as well as the alignment towards RDF in RDFRefine are fully performed through a Web user interface. Unfortunately, RDF Refine does not allow to map CSV input files to RDF. OpenRefine can also handle JSON data but does not implement multi-level iterators that make the mapping complicated when processing deep structures.

YARRRML [13] is a human-readable mapping language developed by the University of Ghent, it allows to define rules and to convert them into RML or R2ML mapping language. Therefore, the RML mapping file generated contains rules to be used to transform input data into an RDF format. The web-based tool named Matey offers the possibility to write YARRML rules in a user-friendly way.

The Matey [15] user interface consists of 4 windows. The top left window contains input data (for example a JSON file). The top right window, the most important one, is the YARRRML editor, where the user can write the YARRRML mapping script. Pushing the "Generate RML" button will automatically translate the YARRRML script to RML and fill in the bottom right windows. Finally, pushing the "Generate LD" button will fill in the bottom left window with the RDF file resulting from the mapping of the input data using RML rules.

The BIGG project focuses on RML as a mapping language for the following technical reasons:

1. RML is based upon RDF that is consistent with the harmonized data.
2. RML mappings are reproducible and maintainable.
3. RML is an extension of a W3C specifications: R2RML [14].
4. RML is not limited to one type of input format and can easily handle JSON, CSV and XML formats. But mostly, in the scope of the BIGG project, it was important that the mapping files could be generated (or at least updated) by non-informatician persons responsible for the different business cases. In this perspective, we have concluded that RML was the best option, thanks to the YARRRML language and Matey tools, which offer the most suitable solution for our BIGG project. We further added a mapping stage based on SPARQL to align URIs controlled values and taxonomies.

IV.2.2. Implementation of mapping by CSTB

For the harmonization of data done by CSTB for the demonstration in Greece (DomX, HELEXIA and HERON data) corresponding to Business Cases 4, 5 and 6 (use cases 8-15), Matey is used to develop and test the mapping rules in YARRRML language, generate RML rules and export them into a file.

The YARRML mapping file is composed of three main parts:

(1) The declaration of prefixes:

Some prefixes and namespaces are predefined by default in YARRRML, and a set of customs ones can be added by adding the 'prefixes' collection to the root of the document. In the example of Figure 14, prefixes for the BIGG Ontology are added.

(2) The declaration of sources:

Under the tag “sources” we list all the “entry points” into the input data. Each entry point is defined by the file name, the format, and an iterator acting as a “regular expression” to filter data. For example:

- Iterator: “$.building[*]” will iterate over all “building” entities in the file
- Iterator: “$..contents[?(@.type=="SITE"&&@.categoryCode=="BUILDING")]” will iterate over all “contents” entities at any level in the JSON tree, and which have specific values of “type” and “categoryCode” properties.

(3) The mapping rules:

Finally, the ‘mappings’ collection contains the declaration of the mapping rules explaining how to translate the current entity pointed by the iterator into triples (subjects, predicates, objects) of the BIGG Ontology. For the example of Figure 14, the mapping named “building” take as an input source the iterator “data1”, and instantiate an object named “$(name)” as a bigg:Building, which has a bigg::name of “$(name)”, and has a relation to a space declared as “space_$(name)”. Instantiation of the bigg::BuildingSpace will be done in the next mapping rule.

```
{
    "building": [
        {
            "name": "BatA",
            "storeys": [
                {
                    "name": "RdC",
                    "spaces": [
                        {"name": "space001"},
                        {"name": "space002"}
                    ]
                },
                {
                    "name": "R+1",
                    "spaces": [
                        {"name": "space101"},
                        {"name": "space104",
                            "devices": [
                                {"name": "ThSensor104-01"},
                                {"name": "HdSensor104-02"}
                            ]
                        }
                    ]
                }
            ]
```

```
prefixes:
 bigg: "http://bigg-project.eu/"
 i: "http://bigg-project.eu/instances/"
 grel: "http://users.ugent.be/~bjdmeest/function/grel.ttl#"

sources:
    data1:
        access: data.json
        referenceFormulation: jsonpath
        iterator: "$.building[*]"
    data1bis:
        access: data.json
        referenceFormulation: jsonpath
        iterator: "$.building[*]"
    data2:
        access: data.json
        referenceFormulation: jsonpath
        iterator: "$.building[*].storeys[*]"
    data3:
        access: data.json
        referenceFormulation: jsonpath
        iterator: "$.building[*].storeys[*].spaces[*]"
    data4:
        access: data.json
        referenceFormulation: jsonpath
        iterator: "$.building[*].storeys[*].spaces[*].devices[*]"

mappings:
  building:
    sources: data1
    s: i:$(name)
    po:
      - [a, bigg:Building]
      - [bigg:name, $(name)]
      - p: bigg:guid
        o:
            function: grel:string_md5
            parameters:
                - [grel:valueParameter, $(name)]
      - p: bigg:hasSpace
        o: i:space_$(name)~iri

  buildingSpace:
    sources: data1bis
    s: i:space_$(name)
    po:
      - [a, bigg:BuildingSpace]
      - [bigg:name, space_$(name)]
      - p: bigg:hasSubSpace
        o: i:space_$(storeys[*].name)~iri
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix bigg: <http://bigg-project.eu/> .
@prefix i: <http://bigg-project.eu/instances/> .
@prefix grel: <http://users.ugent.be/~bjdmeest/function/grel.ttl#> .

i:BatA rdf:type bigg:Building ;
    bigg:name "BatA" ;
    bigg:guid "4c5ec575c534efc17014c19dc6c2a53f" ;
    bigg:hasSpace i:space_BatA .

i:BatB rdf:type bigg:Building ;
    bigg:name "BatB" ;
    bigg:guid "37e5fc91cc8c10cda8397f490b803c45" ;
    bigg:hasSpace i:space_BatB .

i:space_BatA rdf:type bigg:BuildingSpace ;
    bigg:name "space_BatA" ;
    bigg:hasSubSpace i:space_RdC, i:space_R%2B1, i:space_R%2B2 .

i:space_BatB rdf:type bigg:BuildingSpace ;
    bigg:name "space_BatB" .

i:space_RdC rdf:type bigg:BuildingSpace ;
    bigg:name "space_RdC" ;
    bigg:hasSubSpace i:space001, i:space002, i:space003, i:space004 .
```

Figure 14 - Example of a JSON input file (top left) with the corresponding YARRRML mapping (right) and the RDF file created by the mapping (bottom left)

```json
[
  {
    "parentId": null ,
    "assetId": 57 ,
    "name": "BIGG" ,
    "code": "BIGG" ,
    "description": "" ,
    "type": "COMPANY" ,
    "categoryCode": "COMPANY" ,
    "hasChildren": false ,
    "contents": [
      {
        "parentId": 57 ,
        "assetId": 158 ,
        "name": "Cordia" ,
        "code": "Cordia" ,
        "description": null ,
        "type": "GROUP" ,
        "categoryCode": "GROUP" ,
        "hasChildren": false ,
        "contents": [
          {
          {
          {
          {
            "parentId": 158 ,
            "assetId": 58 ,
            "name": "Interamerican" ,
            "code": "Interamerican" ,
            "description": "" ,
            "type": "SITE" ,
            "categoryCode": "OFFICE" ,
            "hasChildren": false ,
            "contents": [
              {
                "parentId": 58 ,
                "assetId": 61 ,
                "name": "HVAC" ,
                "code": "Interamerican-HVAC" ,
                "description": "" ,
                "type": "SITE_ENTITY" ,
                "categoryCode": "ENTITY_GROUP" ,
                "hasChildren": false ,
                "contents": [
                  {
                    "parentId": 61 ,
                    "assetId": 155 ,
```


```yaml
prefixes:
 bigg: "http://bigg-project.eu/"
 i: "http://bigg-project.eu/instances/"
 grel: "http://users.ugent.be/~bjdmeest/function/grel.ttl#"
 energis: "https://energis.cloud/fr/"

sources:
    data_p:
        access: data.json
        referenceFormulation: jsonpath
        iterator:
        "$..contents[?(@.name==\"Cordia\"&&@.type==\"GROUP\"&
        &@.categoryCode==\"GROUP\")]"
    data_b:
        access: data.json
        referenceFormulation: jsonpath
        iterator:
        "$..contents[?(@.name==\"Interamerican\"&&@.type==\"S
        ITE\"&&@.categoryCode==\"OFFICE\")]"
    data_dm:
    data_ts:
    data_ss:

mappings:
  project:
    source: data_p
    s: i:project_$(assetId)
    po:
      - [a, bigg:Project]
      - [bigg:projectName, $(name)]
      - [bigg:projectIdFromOrganization, $(assetId)]
      - p: bigg:affectsBuilding
        o:
          mapping: building
          condition:
            function: equal
            parameters:
              - [str1, $(assetId), s]
              - [str2, $(parentId), o]

  building:
    source: data_b
    s: i:building_$(assetId)
    po:
      - [a, bigg:Building]
      - [bigg:buildingName, $(name)]
      - [bigg:buildingIdFromOrganization, $(assetId)]
```

Figure 15 - Example with real data (HELEXIA / ENERGIS)

IV.2.3. Implementation of mapping by CIMNE

The implementation of mapping for the BIGG Business Cases 1, 2, 3 (use cases 1-7) was done by CIMNE using a custom library implemented in Python. This library, similar to the standard ones, created the transformation of the data into the harmonized RDF format of BIGG. The reason for the implementation of this library was that the data harmonization for these cases was done before the implementation of the Harmonizer component. The example of this is presented below and shows an alternative way for data harmonization, demonstrating the existing versatility of using the BIGG solutions.

The CIMNE transformation tool is less general than the YARRRML proposed by CSTB, which can provide a mapping to any ontology. In the CIMNE case, transformation to RDF is always done to the BIGG ontology and one only needs to specify the class and data fields to apply the transformation.

Inside each JSON mapping one specifies the “mapping” section to create the element, and the “link” section to create the links between the elements. Then the utility will run and create the resulting RDF with the requested data information.

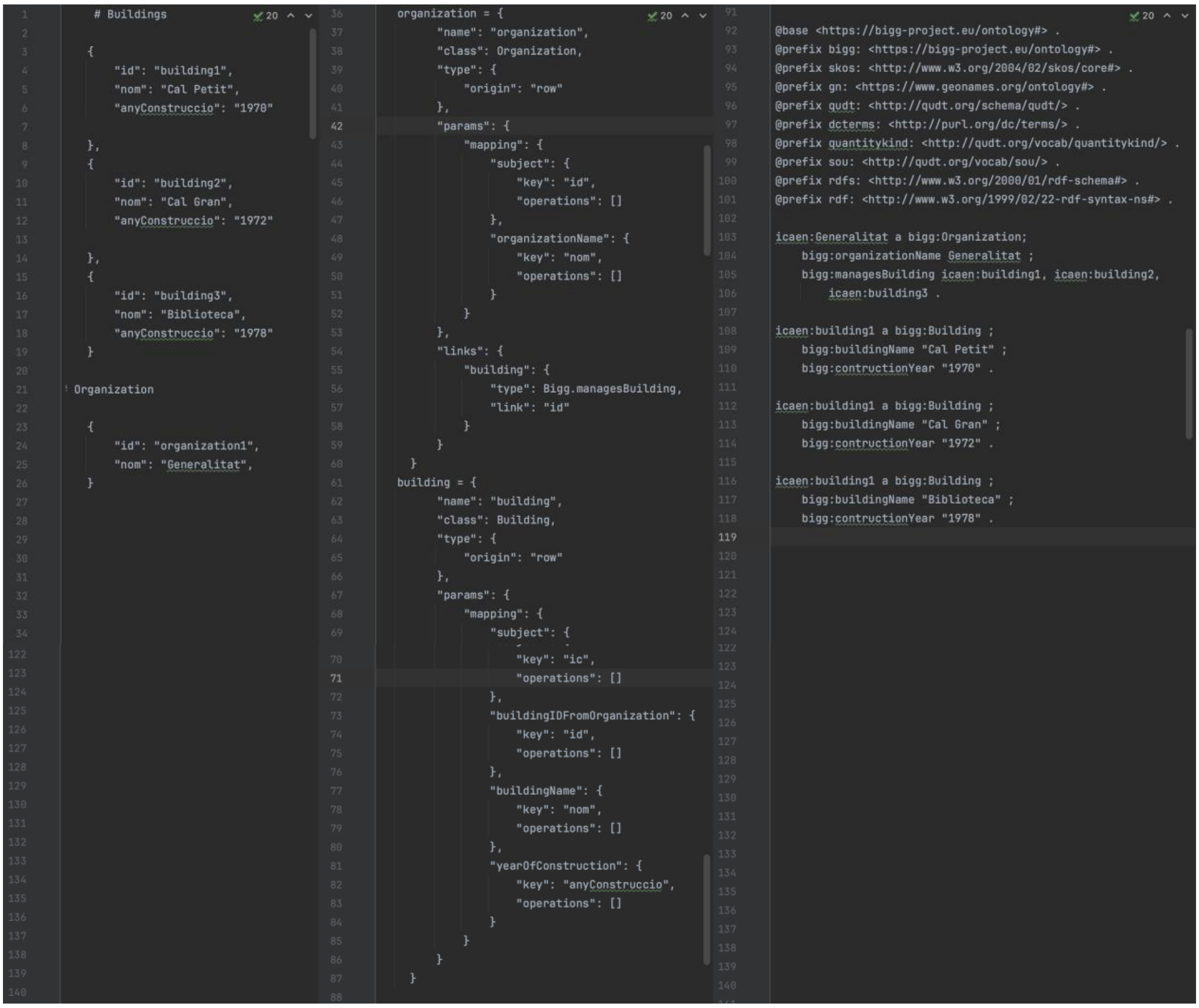


Figure 16 - Mapping of buildings with CIMNE's Tool:

input data (left) mapping (centre) and RDF result (right)

IV.3. Harmonizer functionalities and transformations

The BIGG harmonizer aims at converting any data in the scope of building energy performance, to fit with the BIGG ontology, into RDF. Data that cannot be mapped, or that is not needed for BIGG use cases, is filtered out. As possible input formats in our first approach, we focus on JSON for data coming from sensors (time series) and Asset Management Software (building and systems description) as well as CSV coming from Open Data (geolocation) and RDF (data aligned on SAREF, SOSA or IFC) for alignment with existing ontologies.

The harmonizer must generate data respectful of W3C recommendations and BIGG ontology by providing the following functionalities:

F1: Converts JSON files into RDF compliant with BIGG ontology

F2: Converts CSV files into RDF compliant with BIGG ontology

F3: Aligns Data described using standard ontologies covering the same scope (ifcOWL, SAREF, SSN/SOSA, GeoNames, QUDT, WGS84, FOAF)

F4: Allows to map input objects with BIGG classes

F5: Allows to map input attributes with BIGG data properties

F6: Interprets implicit links between objects through object properties.

F7: Reconciles input values with open registers, BIGG taxonomies and enumerations

F8: Materializes data context and generate 5-stars data

The following parts describes how the harmonizer provides F1-F8 functionalities.

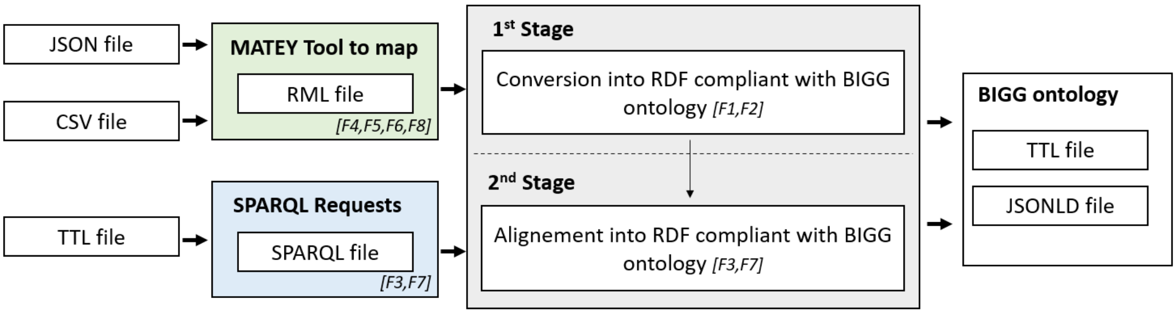


Figure 17 - General workflow of the Harmonizer and its functionalities

Converting input formats to RDF

BIGG is intended to address the necessities of Digital Building Twins for high-quality data, known as 5-star data, as originally introduced by Tim Berners-Lee. Frequently, the raw data obtained by a Digital Building Twin needs cleaning and enrichment to enable full interoperability. The levels established by Tim Berners-Lee consist of five stages: availability (level 1), structuring (level 2), openness (level 3), universal identification (level 4), and linkage to other data (level 5). In our case, raw data is mainly provided as JSON, but the same methodology could be applied to CSV-formatted data. In this project, the 5-star level is achieved by following the RDF specifications.

The first stage of the harmonizer extracted inputs from a JSON file and a RML mapping file to produce a first RDF file, matching the BIGG scope. The RML mapping files are composed of iterators to identify data sources (using the JSON path syntax) and class mapping declarations that refer to iterators. Mapping files can either be produced manually or by using dedicated tools that simplify the mapping process, e.g., RMLEditor or Matey.

RML processing is based on the RMLMapper library developed by Ghent University. First, the relevant parts of the JSON are identified according to iterators declared in the mapping file. Then the mapping itself is performed by creating, for each mapping data, an instance of the mapped ontology class by generating a URI and by generating triples according to data properties. Each URI is generated from the context (provider or building owner and building) and the local ID defined in the JSON. The context is used to generate the prefix. The class name followed by the local id are concatenated with the prefix. At this step, the literal values are not yet converted.

Interpreting implicit link

The underlying structures of JSON are arrays and trees. Relations between objects are implicitly declared by using the native parent-child relation provided by the tree structure. As BIGG ontology defines several relations, the harmonized need to match the parent-child relation between two JSON objects with one of the BIGG ontology relation. Figure 18 illustrates how two parent-child relation from the same JSON file can be interpreted as distinct relations in the BIGG ontology.

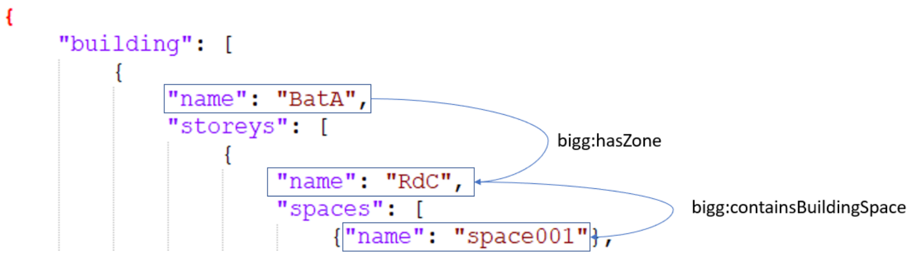


Figure 18 - Sample of JSON file describing building structure

The materialization of implicit links is also ensured by the harmonizer’s first mapping stage. Object properties (as declared in the mapping file) map instances with each other’s by using their URI. As for instance generation, JSON samples corresponding to the subject and the object of a triple are identified using absolute or relative JSONPath expressions. As URI are always generated by using the same convention, URIs remain consistent. This way, URIs generated when populating instances match with URIs generated when building relations between instances.

Reconciliating data with enumerations

The BIGG Harmonizer uses GeoNames as a universal register for public buildings. By reconciling GeoNames buildings with BIGG building descriptions, the project allows for the federation of data provided by building owners with data provided by the Open Data community. This is achieved by querying the GeoNames endpoint to retrieve the building's public URI, official name, and address, using the building location as a reference. The harmonizer's first stage generates a graph, which is then used to execute a federated SPARQL query in the second stage to replace the temporary building URI with the GeoNames URI and insert complementary information requested from GeoNames. Additionally, this project proposes the use of alternate public endpoints, such as Open Street Map's endpoint, to provide data on building height, roof inclination, and envelope materials.

Figure 19 - GeoNames providing universal ID and description for public buildings

BIGG ontology is a valuable resource that provides taxonomies for the classification of various objects, including buildings, spaces, and sensors. Within the context of building usage classification, an excerpt of the building usages taxonomy is presented in Figure 20, which outlines the primary services provided by a building.

Property	Taxonomy 1st level	Taxonomy 2nd level
buildingSpaceUseType	EducationAndResearch	ExtracurricularEducationCenter
		Laboratory
		MilitaryOrPoliceAcademy
		OtherEducationAndResearch
		Preschool
		PrimarySchool
		ResearchCenter
		SecondarySchool
		University
		(empty)
	Healthcare	DayCareHospital
		HomelessShelter
		Hospital
		Mortuary
		NursingHome

Figure 20 - Sample of one of the BIGG taxonomies classifying building space usages.

In situations where the classification of raw data differs from the BIGG classification, the harmonizer's second stage is used to translate between the two. For example, the French building classification term "Université" must be converted to its corresponding term in the BIGG taxonomy, namely "University". A thesaurus is embedded in the BIGG ontology. It uses the SKOS ontology to describe taxonomies (skos:scheme), translations (rdf:label "Université"@fr, "University"@en) and cross-taxonomy semantic links (<medical center> skos:match <hospital>). In the harmonizer's second stage, the reconciliation process is accomplished through a federated query that uses the thesaurus to translate from the taxonomy used by the raw data to the harmonized taxonomy. Once the corresponding term is located, the query substitutes the appropriate value in the graph that was previously created by the harmonizer's initial stage.

Making explicit the data context

When describing energy consumption as JSON time series, the contextual information regarding the source (Who), location (Where), and method (How) of measurement is often overlooked or indirectly specified in the file name. A standardized approach is required to extract and retrieve contextual information, such as the location, building, or feature of interest, from the pre-existing database or other data sources. The identification of data providers and building owners can be incorporated into the mapping file (in the SPARQL mapping), which can be further used by the harmonizer to enhance the graph. In instances where the timestamp is absent, the current date and time can be used to trace the incoming data. As an additional improvement, the harmonizer can also extract pertinent data, such as building and sensor identification, from the file name, to further augment the graph.

Aligning data

In addition to the raw data collected as a JSON file, the harmonizer enables the ingestion of RDF data that implements standardized ontologies.

The BIGG ontology has been aligned with several existing ontologies related to Digital Building Twins, such as BOT (for building topology description), SAREF (for energy system description), and SSN/SOSA (for sensor and time series description).

This alignment consists of triples that declare corresponding classes (owl:equivalentClass) and corresponding predicates (owl:equivalentProperty) from one ontology to the other. The data alignment is generally performed on a given opportunity via a semantic reasoner embedded in the triple store but, if necessary, the harmonizer can materialize class and properties alignments by substituting the original vocabulary with BIGG vocabulary.

There are two ways to extend the harmonizer's alignment capabilities. The first way is to extend the ontology with new class and property equivalences. The second way is to add SPARQL mapping queries that will convert classes and properties from one vocabulary to another (the 2nd mapping stage in Figure 16). Whereas the first way appears to be the most standard, the second solution provides more flexibility as benefits from the expressiveness of the SPARQL language to perform complex transformations such as aggregations.

IV.4. Public documentation of the Harmonizer

The BIGG Harmonizer documentation is available on the public GitHub repository of the BIGG project: https://github.com/biggproject/Harmonizer/tree/main

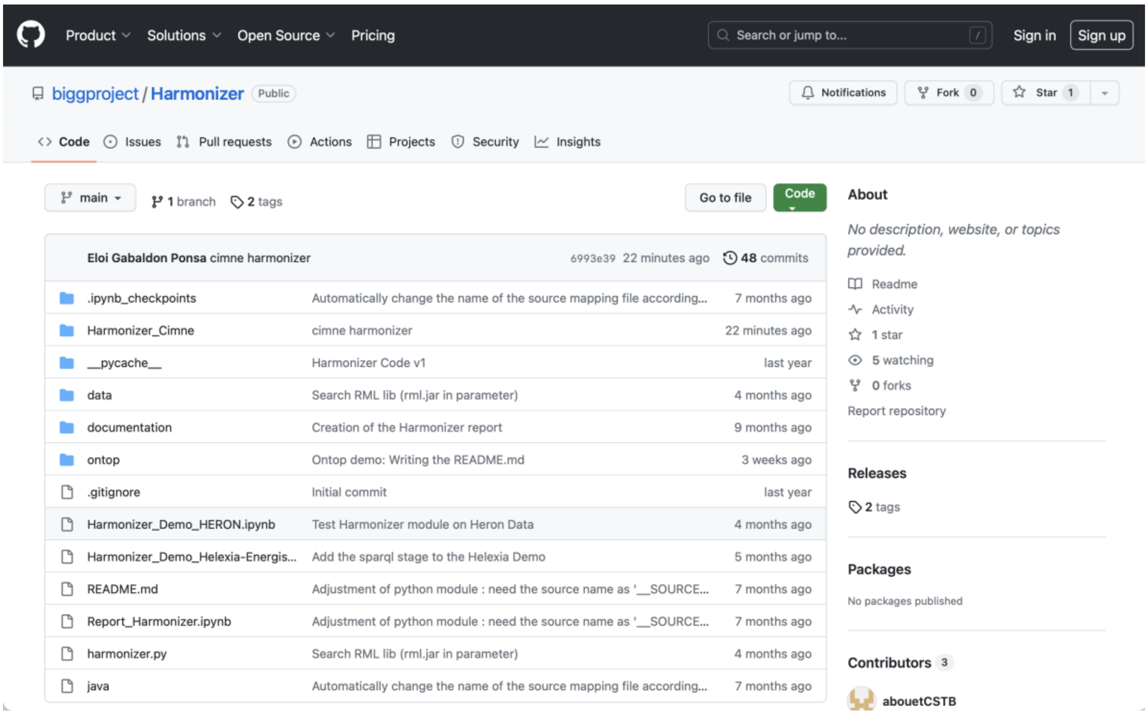


Figure 21 - BIGG Harmonizer documentation on GitHub

V. BIGG STANDARDS-BASED ONTOLOGY (BIGGSTD)

One of the objectives of BIGG project is the technical contribution to EU standards and ontologies (Task 4.4) by proposing additions and extensions that would enable them to support the wide scope of use cases for big data in buildings covered by BIGG. This task was approached by working on maximally reusing existing standards in the BIGG ontology, thus evidencing the specific contributions that BIGG could offer in terms of concepts and relations that do not have any equivalent in the current standards. As a result, a set of classes, data properties, and object properties were proposed by BIGG in Annex 1, to be considered in the further extension of the standards, which would allow them to support the BIGG use dases.

The task required a significant effort to search for existing ontologies and to analyze their concepts in the context of use in BIGG. This work finally led to transforming the BIGG ontology into the standards-based ontology BIGGstd. This section provides an overview of the process of development and the resulting ontology. The full documentation of BIGGstd is provided on the public repository of the BIGG project on GitHub: https://github.com/biggproject/Ontology/.

Methodology

The BIGG ontology presented in Section III has been used as a conceptual base for the development of BIGGstd. The advantage of this approach is the previous validation and assurance of applicability as the BIGG ontology has been already tested with the use cases in the demonstrations.

The development process of the BIGGstd ontology could be summarised in the following steps:

(i) search and analysis of concepts from existing ontologies;

(ii) modelling: developing the relationships between concepts;

(iii) generation of the ontology in TTL (Turtle) format.

The search of terms in existing ontologies was performed with the help of the Linked Open Vocabularies tool and direct access to the ontologies' documentation:

https://lov.linkeddata.es/dataset/lov/

The modeling was performed by developing UML class diagrams representing the relationships between classes. The final generation of the ontology was done with the help of the Chowlk framework, which generated the ontology in TTL format.

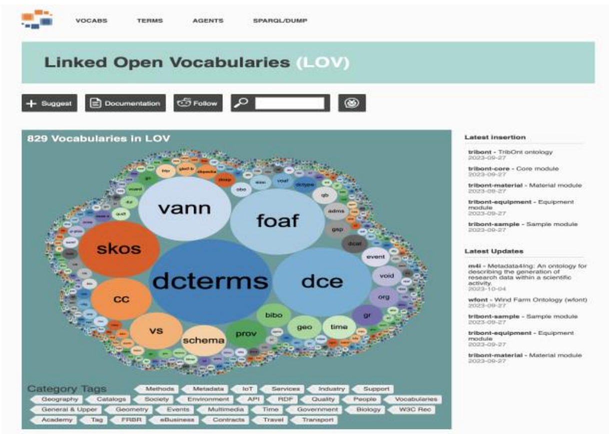


Figure 22 - Linked Open Vocabularies tool.

V.1. Overview of BIGGstd ontology

The BIGGstd ontology is developed as modular ontology consisting of core and extensions, each of which has separate extensions allowing further evolution in the future:

- bigg:http//:bigg-project.eu/Id/ontology#
- bigg4kpi:http//:bigg-project.eu/Id/bigg4kpi#
- bigg4co2:http//:bigg-project.eu/Id/bigg4co2#
- bigg4tariff:http//:bigg-project.eu/Id/bigg4tariff#
- bigg4proj:http//:bigg-project.eu/Id/bigg4proj#

Each of the BIGGstd ontology's parts has extensively reused concepts from existing ontologies and added additional concepts that are necessary to support the BIGG use cases.

The overall structure of the BIGGstd is presented in the Figure 23 below and the detailed structure of each part is presented in continuation. The full documentation of the ontology is provided in the GitHub repository if the project.

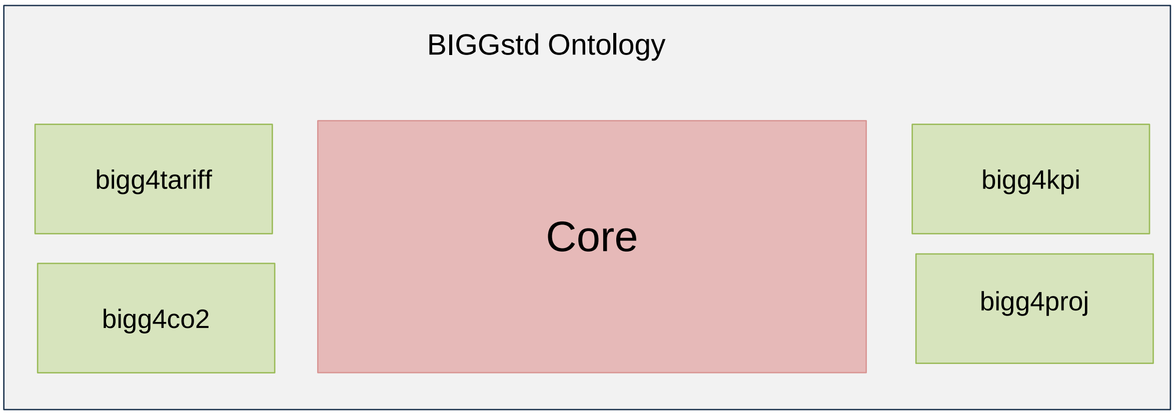


Figure 23 - Structure of the BIGGstd ontology

The core of BIGGstd encompasses the main classes that are necessary to allocate the general information for all the use cases covered by the BIGG project. The extensions contain parts of the ontology that are only relevant for some of the use cases. The reused classes in the ontology are represented through white boxes, while the particular classes introduced by the BIGG project are highlighted in blue. The modules of the ontology are presented below in a structured way through class diagrams and namespaces definitions used in the ontology.

The specific contribution of BIGG as proposed classes, data properties and object properties that might represent interest for the evolvement of standards are presented in Annex 1.

Core

The core classes in BIGGstd and their relations provide the general structure for allocating and interconnecting the building data relevant to the BIGG use cases. This includes the general building characteristics, building systems, components, devices, as well as the relation of the building to the wider context of geospatial location within the urban environment, its use, and responsible persons for its management.

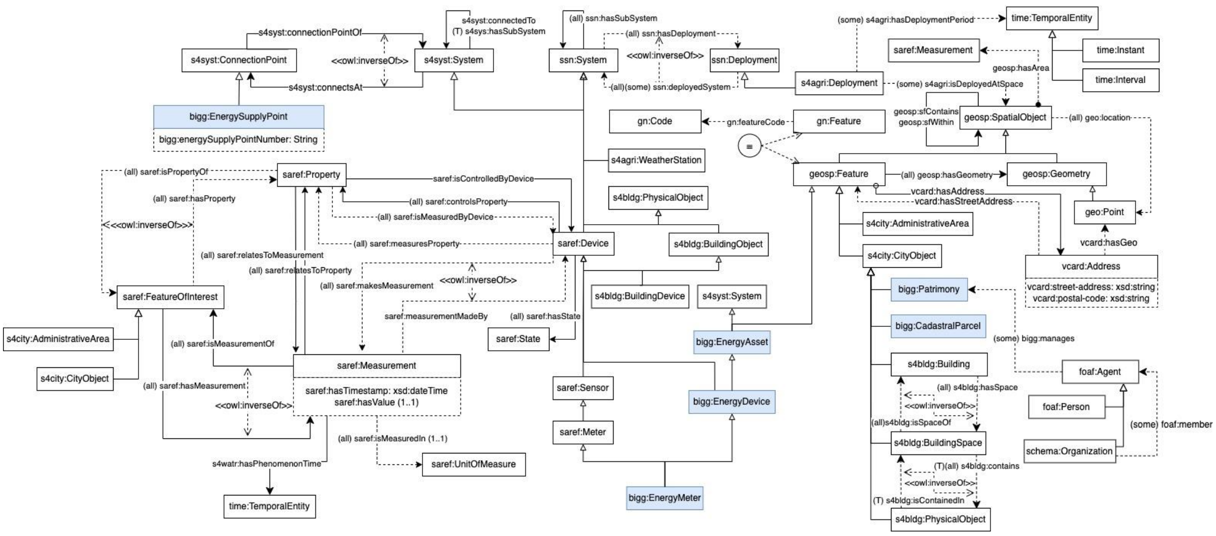


Figure 24 - BIGGstd ontology core

bigg: http://bigg-project.eu/ld/ontology#
foaf: http://xmlns.com/foaf/0.1/
schema: https://schema.org/
geo: http://www.w3.org/2003/01/geo/wgs84_pos#
geosp: http://www.opengis.net/ont/geosparql#
gn: http://www.geonames.org/ontology#
saref: https://saref.etsi.org/core/
s4agri: https://saref.etsi.org/saref4agri/
s4syst: https://saref.etsi.org/saref4syst/
s4city: https://saref.etsi.org/saref4city/
s4bldg: https://saref.etsi.org/saref4bldg/
ssn: http://www.w3.org/ns/ssn/
time: http://www.w3.org/2006/time#

dc:creator: Edgar Martínez-Sarmiento
dc:creator: ...
owl:versionInfo: 2.1
dc:title: BIGG Ontology

Extension bigg4kpi

This extension was created for allocating the different results produced in the context of the BIGG use cases. It reuses some general classes from existing ontologies and adds classes and relations that are particular to the results of the BIGG project.

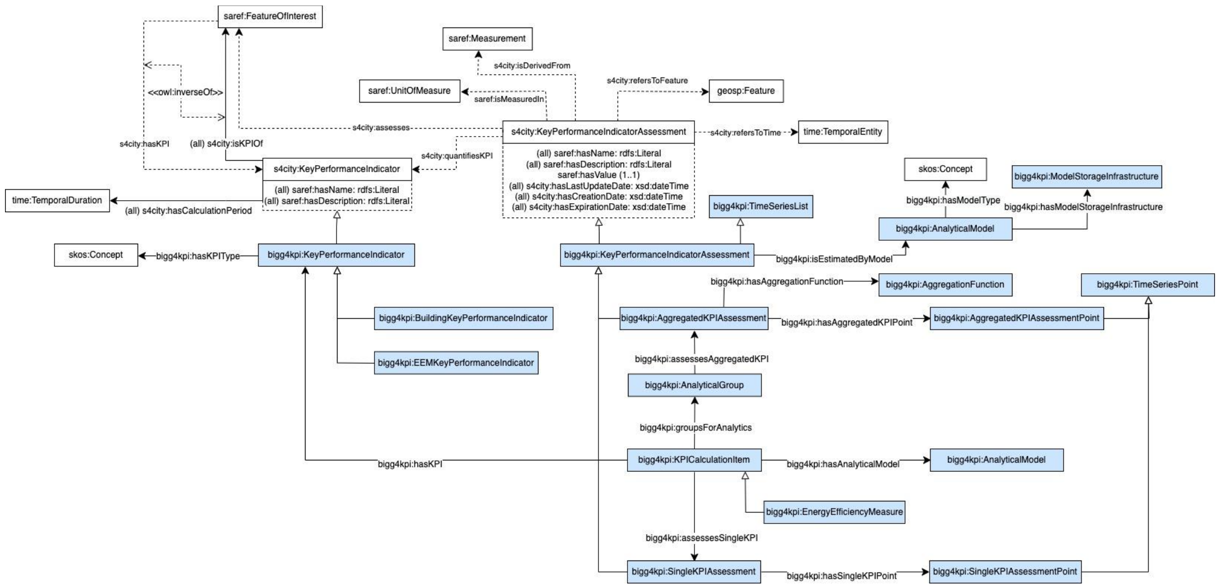


Figure 25 - BIGGstd ontology extension bigg4kpi

bigg4kpi: http://bigg-project.eu/ld/bigg4kpi#
bigg: http://bigg-project.eu/ld/ontology#
saref: https://saref.etsi.org/core/
s4city: https://saref.etsi.org/saref4city/
geosp: http://www.opengis.net/ont/geosparql#
skos: http://www.w3.org/2004/02/skos/core#
time: http://www.w3.org/2006/time#
dcterms: http://purl.org/dc/elements/1.1/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs: http://www.w3.org/2000/01/rdf-schema#
owl: http://www.w3.org/2002/07/owl#
xsd: http://www.w3.org/2001/XMLSchema#

dcterms:creator: Edgar Martínez-Sarmiento
dcterms:contributor: Eloi Gabaldon
dcterms:contributor: Guillaume Picinbono
dcterms:contributor: Nicolas Bus
dcterms:contributor: Stoyan Danov
dcterms:contributor: Jordi Carbonell
owl:versionInfo: 1.0.0
dcterms:title: The BIGG for Key Performance Indicators ontology extension

Extension bigg4co2

This extension defines classes and relationships that are necessary for allocating the results related to the emissions from energy use.

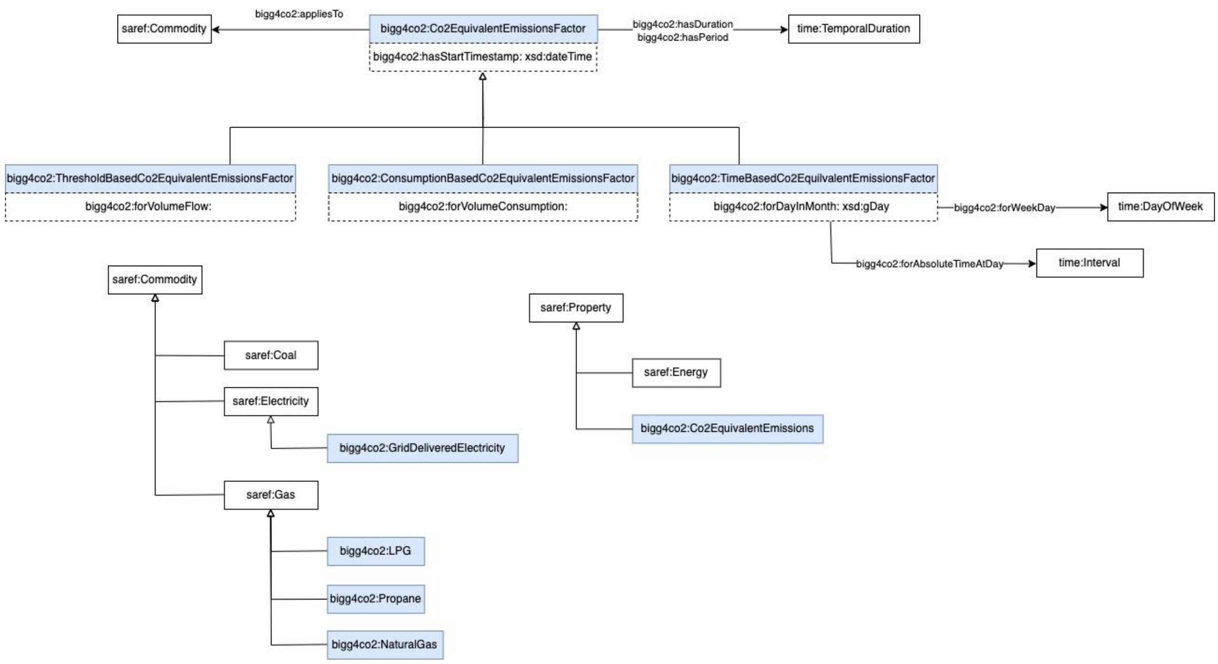


Figure 26 - BIGGstd ontology extension bigg4co2

bigg4co2: http://bigg-project.eu/ld/bigg4co2#
bigg: http://bigg-project.eu/ld/ontology#
saref: https://saref.etsi.org/core/
time: http://www.w3.org/2006/time#
dcterms: http://purl.org/dc/terms/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs: http://www.w3.org/2000/01/rdf-schema#
owl: http://www.w3.org/2002/07/owl#
xsd: http://www.w3.org/2001/XMLSchema#

dcterms:creator: Edgar Martínez-Sarmiento
dcterms:contributor: Eloi Gabaldon
dcterms:contributor: Guillaume Picinbono
dcterms:contributor: Nicolas Bus
dcterms:contributor: Stoyan Danov
dcterms:contributor: Jordi Carbonell
owl:versionInfo: 1.0.0
dcterms:title: The BIGG for CO2 emissions ontology extension

Extension bigg4tariff

The extension "tariff" was created to represent information about different tariffs of utilities delivered to buildings, such as energy and water.

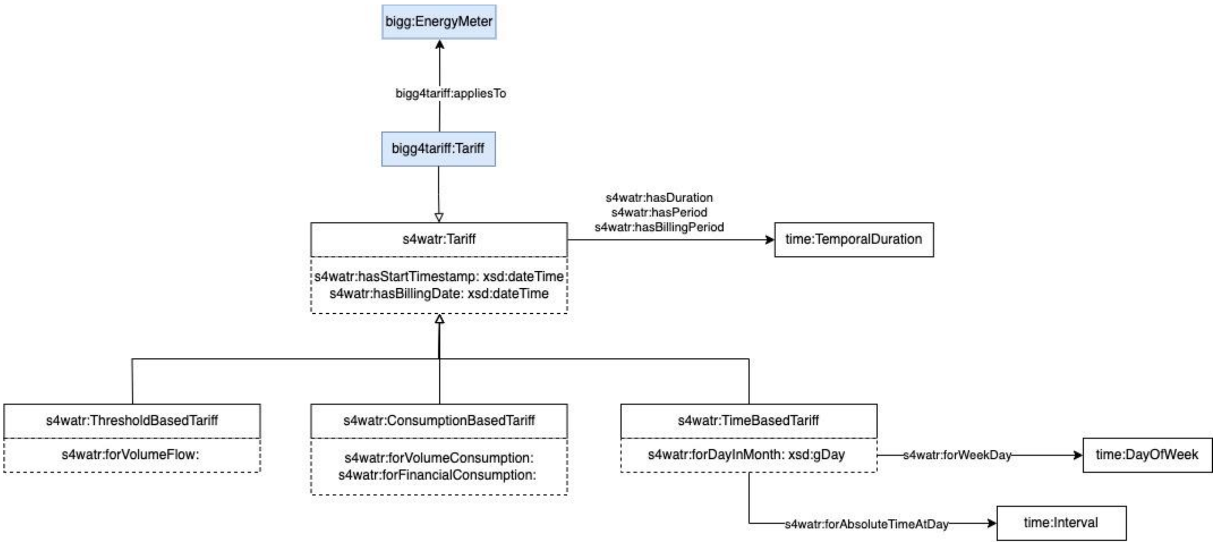


Figure 27 - BIGGstd ontology extension bigg4tariff

bigg4tariff: http://bigg-project.eu/ld/bigg4tariff#
bigg: http://bigg-project.eu/ld/ontology#
saref: https://saref.etsi.org/core/
s4watr: https://saref.etsi.org/saref4watr/
time: http://www.w3.org/2006/time#
dcterms: http://purl.org/dc/elements/1.1/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs: http://www.w3.org/2000/01/rdf-schema#
owl: http://www.w3.org/2002/07/owl#
xsd: http://www.w3.org/2001/XMLSchema#

dcterms:creator: Edgar Martínez-Sarmiento
dcterms:contributor: Eloi Gabaldon
dcterms:contributor: Guillaume Picinbono
dcterms:contributor: Nicolas Bus
dcterms:contributor: Stoyan Danov
dcterms:contributor: Jordi Carbonell
owl:versionInfo: 1.0.0
dcterms:title: The BIGG for Tariff ontology extension

Extension bigg4proj

This extension defines different classes of projects related to buildings in the context of BIGG. These include building renovation, retrofitting and energy performance contracts as well as related concepts.

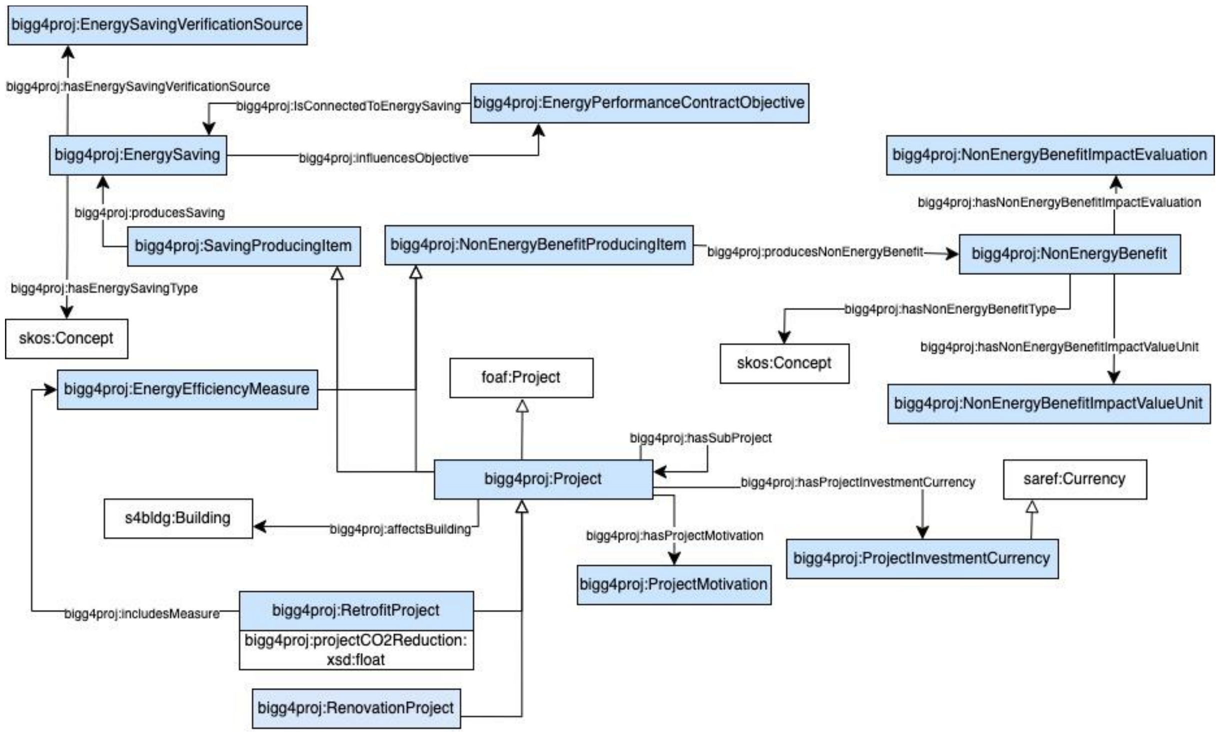


Figure 28 - BIGGstd ontology extension bigg4proj

bigg4proj:http://bigg-project.eu/ld/bigg4proj#
bigg:http://bigg-project.eu/ld/ontology#
foaf: http://xmlns.com/foaf/0.1/
saref: https://saref.etsi.org/core/
s4bldg: https://saref.etsi.org/saref4bldg/
skos: http://www.w3.org/2004/02/skos/core#
dcterms: http://purl.org/dc/elements/1.1/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs: http://www.w3.org/2000/01/rdf-schema#
owl: http://www.w3.org/2002/07/owl#
xsd: http://www.w3.org/2001/XMLSchema#

dcterms:creator: Edgar Martínez-Sarmiento
dcterms:contributor: Eloi Gabaldon
dcterms:contributor: Guillaume Picinbono
dcterms:contributor: Nicolas Bus
dcterms:contributor: Stoyan Danov
dcterms:contributor: Jordi Carbonell
owl:versionInfo: 1.0.0
dcterms:title: The BIGG for projects ontology extension

V.2. Public documentation

The BIGGstd standards-based ontology and its full documentation is available on the public GitHub repository of the BIGG project: https://github.com/biggproject/Ontology/tree/main/BIGGstd

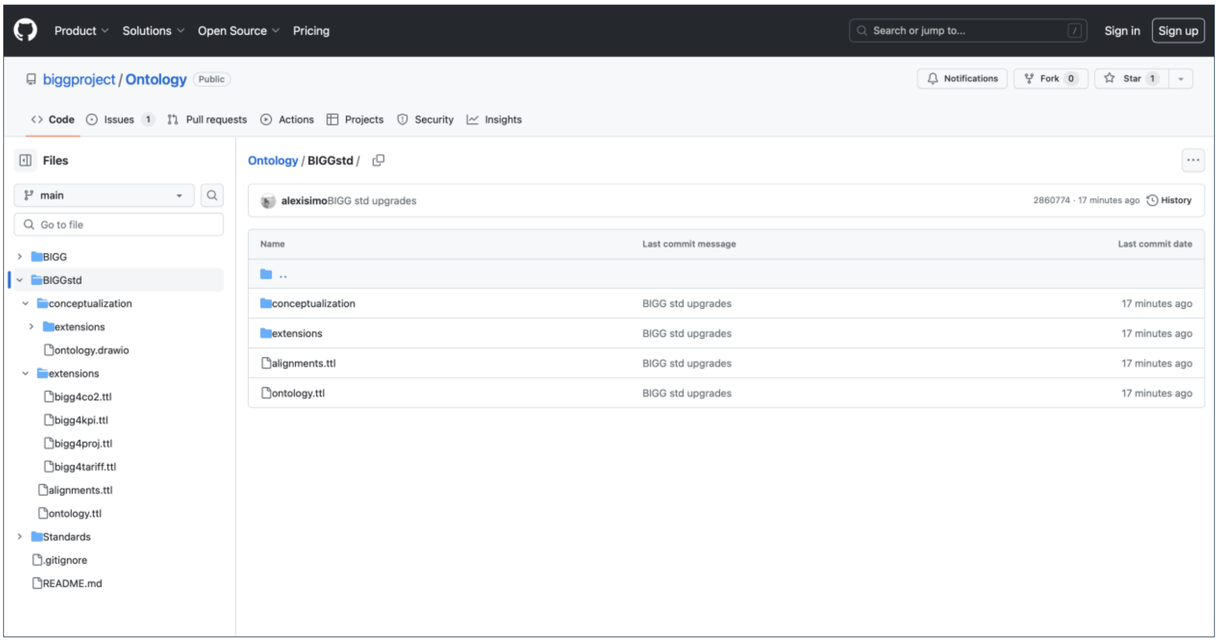


Figure 29 - BIGGstd ontology documentation on GitHub

VI. CONCLUSIONS AND FUTURE WORK

The BIGG harmonization layer is a core component in the BIGG Reference Architecture Framework that ensures its capability to align, harmonize, and make comparable data from different sources. The data harmonization makes possible the use of the BIGG AI Toolbox in the implementation of the BIGG use cases, with the harmonized data format being also key for the overall communication and integration of the IT architecture's components.

The final harmonization layer consists of a BIGG ontology for internal data representation, BIGG Harmonizer tools, and mappings to external data sources and sinks that enable the semantic interoperability of the BIGG solution.

The development of the harmonization layer was done in three steps. The first step was the development of the BIGG Standard Data Model 4 Buildings, based on a detailed analysis of the requirements of the BIGG use cases, and of the datasets available from the BIGG pilots. The data model set the semantic base and structure of data in BIGG and served as a common reference for the parallel work in the work packages on communication (WP3), AI toolbox development (WP5), and system integration (WP2). The data model is reported in deliverable D4.1.

The second step was the development of the BIGG Ontology, a W3C-standards-compliant ontology based on RDF specifications enabling the use of semantic technologies and machine understanding of data. The ontology was developed on the conceptual base of the BIGG Standard Data Model 4 Buildings and followed the naming conventions provided by it to facilitate the collaborative work in the project and the continuity of the work from the previous phase. In parallel, this step developed the BIGG Harmonizer. This component enabled the automated harmonization of data, facilitated by the adoption of the RDF format. The Harmonizer orchestrates the transformation of heterogeneous data to RDF, covering building and sensors description as well as energy -related time series. It is fully compliant with W3C recommendations by composing an RML transformation with SPARQL transformations. The Harmonizer solution is extensible to any data provider. By providing a mapping file, it creates harmonized data streams enabling the use of the BIGG AI Toolbox. Along with the Harmonizer, ad-hoc data mapping and harmonization tools have been developed and exemplified, thus providing an alternative for implementation that is feasible for relatively small datasets and for developers familiar with RDF and Python who would like to adopt the BIGG AI Toolbox components.

As a final step, with the objective to contribute to standardization, the BIGG ontology was transformed into a standards-based ontology BIGGstd, by reusing existing standards. The process was iterative and comprised a detailed analysis of existing ontologies, selection of equivalent terms, and reasoning on their adoption in BIGG. The process resulted in the creation of the BIGGstd ontology and served to evidence gaps in existing ontologies and to pinpoint potential contributions to them. The specific classes, data properties and object properties added by BIGG, for which no equivalent terms in existing standards were found, are described in Annex 1. The potential use of these terms for future extensions of existing data standards will be considered by the BIGG representatives in the EU standardisation committees.

VII. References

[1]. Energy Efficiency – the first fuel for the EU Economy: How to drive new finance for energy efficiency investments, Final Report covering Buildings, Industry and SMEs, EEFIG, February 2015, ISBN: 978-84-606-6087-3.

[2]. Building Energy Performance Gap Issues: An International Review. IPEEC Building Energy Efficiency Task Group, 2019 https://www.energy.gov.au/sites/default/files/the_building_energy_performance_gap-an_international_review-december_2019.pdf

[3]. IEA EBC Annex 53: Total Energy Use in Buildings – Analysis and Evaluation Methods. https://www.iea-ebc.org/Data/publications/EBC_PSR_Annex53.pdf

[4]. Pritoni, M., et al., 2021. Metadata Schemas and Ontologies for Building Energy Applications: A Critical Review and Use Case Analysis. Energies, 14(7).

[5]. Cuenca, J., Larrinaga, F. & Curry, E., 2017. A Unified Semantic Ontology for Energy Management Applications. Vienna, s.n.

[6]. Bergman, M. K., 2018. A Knowledge Representation Practionary. Coralville: Springer. https://doi.org/10.1007/978-3-319-98092-8

[7]. Esnaola-Gonzalez, I., Bermúdez, J., Fernandez, I. & Arnaiz, A., 2020. Ontologies for Observations and Actuations in Buildings: A Survey. Semantic Web, 11(4), pp. 593-621

[8]. E. Iglesias, S. Jozashoori, D. Chaves-Fraga, D. Collarana, and M.E. Vidal. SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs

[9]. A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In Proceedings of the Workshop on Linked Data on the Web, co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, 2014.

[10]. Maxime Lefrançois, Antoine Zimmermann, Noorani Bakerally. SPARQL-Generate. A SPARQL extension for generating RDF from heterogeneous formats.

[11]. W3C Recommendation 21 March 2013. SPARQL. https://www.w3.org/TR/sparql11-query/

[12]. Per Larsson. Evaluation of Open Source Data Cleaning Tools: Open Refine and Data Wrangler. http://openrefine.org/

[13]. Ben De Meester, Pieter Heyvaert, Ruben Verborgh, and Anastasia Dimou. Mapping Language Analysis of Comparative Characteristics. https://rml.io/yarrrml/spec/https://rml.io/yarrrml/spec/

[14]. S. Das, S. Sundara, R. Cyganiak, R2RML: RDB to RDF Mapping Language, (2012). https://www.w3.org/TR/r2rml/

[15]. Dylan Van Assche , Thomas Delva , Pieter Heyvaert , Ben De Meester , and Anastasia Dimou. Towards a more human-friendly knowledge graph generation & publication. International Semantic Web Conference (ISWC) 2021: Posters, Demos, and Industry Tracks, https://rml.io/yarrrml/matey/

VIII. ANNEX 1

This section presents the list of concepts of the BIGGstd ontology that have no equivalents in existing ontologies.

VIII.1. core

concepts	type	comment
http://bigg-project.eu/ld/ontology#	http://www.w3.org/2002/07/ow	Ontology
manages	ObjectProperty	Relation between an agent and a patrimony they manage.
energySupplyPointNumber	DatatypeProperty	Number of energy supply point from utility.
EnergySupplyPoint	Class	Energy supply point from utility.
nationalCadastralReference	DatatypeProperty	Thematic identifier at national level, generally the full national code of the basic property unit. Must ensure the link to the national cadastral register or equivalent. (https://www.w3.org/2015/03/inspire/cp)
CadastralParcel	Class	Areas defined by cadastral registers or equivalent. (https://www.w3.org/2015/03/inspire/cp)
EnergyAsset	Class	An energy asset is a physical entity used in the process of transforming, transporting, storing and distributing energy.
EnergyDevice	Class	An energy device is a device that is also an energy asset.
EnergyMeter	Class	An energy meter is an instrument intended to measure continuously, memorize, and display the energy consumption.
Patrimony	Class	A collection of real state entities used for the scope of energy accounting which can contain buildings and parcels.
SurfaceArea	Class	A saref:Property related to some measurements that are characterized by a certain value measured in an surface area unit (such as square meters).
SurfaceAreaUnit	Class	The unit of measure for surface area.
GrossFloorArea	Class	Sum of the floor areas of all building spaces measured from the external faces of the exterior walls.

VIII.2. bigg4tariff

concepts	type	comment
http://bigg-project.eu/ld/bigg4tariff#	http://www.w3.org/2002/07/owl#	Ontology
appliesTo	ObjectProperty	The energy meter to which a tariff applies to.
Tariff	Class	The specifications of a tariff associated with one of the metered commodities.
EUR-PER-MegaW-HR	NamedIndividual	An instance of currency related to energy consumption in Euros per MWh

VIII.3. bigg4proj

concepts	type	comment
http://bigg-project.eu/ld/bigg4proj#	http://www.w3.org/2002/07/owl	Ontology
isConnectedToEnergySaving	ObjectProperty	Relation between an energy performance contract objective and the energy savings it produces.
EnergyPerformanceContractObjective	Class	The energy saving objective of the energy performance contract
EnergySaving	Class	Any estimate or measure of energy savings triggered by a Retrofit Project or Energy Efficiency Measure
RenovationProject	Class	A project that affects a whole Building or part of it, and can indirectly have an effect on the energy efficiency.
Project	Class	A building energy efficiency related project
affectsBuilding	ObjectProperty	Relation that points which buildings are affected by a project
energySavingEndDate	DatatypeProperty	Final date of the reference period for the energy savings
energySavingIndependentlyVerified	DatatypeProperty	Indication of whether the presented energy savings were independently verified or not
energySavingStartDate	DatatypeProperty	Initial date of the reference period for the energy savings
energySavingValue	DatatypeProperty	Energy savings value
geometrySRID	DatatypeProperty	International spatial Reference System for geometric types
hasEnergySavingType	ObjectProperty	Links an energy saving with its type given as a skos concept
hasEnergySavingVerificationSource	ObjectProperty	Relates energy saving to an energy saving verificaion source
EnergySavingVerificationSource	Class	Source of the verification, if the presented energy savings were independently verified
hasNonEnergyBenefitImpactEvaluation	ObjectProperty	Relation existing between a non energy benefit with its impact evaluation
NonEnergyBenefit	Class	Any additional benefit produced by Renovation Projects and Energy Efficiecy Measures other than Energy Savings
NonEnergyBenefitImpactEvaluation	Class	Evaluation of the project impact over the selected non energy benefit
hasNonEnergyBenefitImpactValueUnit	ObjectProperty	Vinculates a non energy benefit impact with an impact value unit
NonEnergyBenefitImpactValueUnit	Class	Units of the approximate net value of the impact of the project over the non-energy benefit
hasNonEnergyBenefitType	ObjectProperty	Links a non energy benefit with its type given as a skos concept
hasProjectInvestmentCurrency	ObjectProperty	Relation between a project and the project investment currency
ProjectInvestmentCurrency	Class	Original currency of the project investment
hasProjectMotivation	ObjectProperty	Links a project with its motivation
ProjectMotivation	Class	Key reasons for the investment
hasSubProject	ObjectProperty	Links a project to its super project.
includesMeasure	ObjectProperty	Relation between a retrofit project and an energy efficiency measure
RetrofitProject	Class	Any retrofit project that affects a whole Building or part of it, and that consists of one or more EnergyEfficiencyMeasures.
EnergyEfficiencyMeasure	Class	Any measure for the improvement of the efficiency of a Building or its Elements.
influencesObjective	ObjectProperty	Relation between an energy saving and an energy performance contract objective
nonEnergyBenefitImpactValue	DatatypeProperty	Approximate net value of the impact of the project over the non-energy benefit
nonEnergyBenefitImpactValueDescription	DatatypeProperty	Description of the value provided for the impact of the project over the non-energy benefit
nonEnergyBenefitImpactValueVerifiedAndMeasured	DatatypeProperty	Indication of whether the impact over the non-energy benefit has been measured and verifed
nonEnergyBenefitImpactVerificationMethod	DatatypeProperty	Description of the verification/measurment method used, in case the impact over the non-energy benefit has been verified or measured
producesNonEnergyBenefit	ObjectProperty	Links a non energy benefit producing item with its non energy benefit
NonEnergyBenefitProducingItem	Class	Any item that is able to produce non energy benefits
producesSaving	ObjectProperty	Relation between a saving producing item and an energy saving
SavingProducingItem	Class	Any item that is able to produce energy savings
projectCO2Reduction	DatatypeProperty	Annual reduction of CO2 emissions by the project, tCO2/year
projectCurrencyExchangeRate	DatatypeProperty	Exchange rate between the original investment currency and euros

projectDescription	DatatypeProperty	Free text for the user to describe the project (e.g 300 characters)
projectDiscountRate	DatatypeProperty	Discount rate used to calculate the financial metrics for the renovation project
projectGrantsShareOfCosts	DatatypeProperty	Estimated share of the total project costs that were covered with grant funding, in case the project received it
projectIDFromOrganization	DatatypeProperty	Project identifier provided by the organisation
projectIncludedConfortmeterSurvey	DatatypeProperty	Indication of whether the project included a Comfortmeter survey
projectIncludedNonEnergyBenefitsEstimate	DatatypeProperty	Indication of whether the non energy benefits produced by the project were estimated
projectInterestRate	DatatypeProperty	Interest rate used to calculate the financial metrics for the renovation project
projectInternalRateOfReturn	DatatypeProperty	Estimated Internal Rate of Return (IRR) of the renovation project
projectInventivesShareOfRevenues	DatatypeProperty	Estimated share of the total project revenues that are represented by incentives schemes, in case the project benefitted/will benefit from them
projectInvestment	DatatypeProperty	Investment for the project inplementation
projectName	DatatypeProperty	Name of the project
projectNetPresentValue	DatatypeProperty	Estimated Net Present Value (NPV) of the renovation project
projectOperationalDate	DatatypeProperty	Date on which the project became operational
projectReceivedGrantFounding	DatatypeProperty	Yes or no data field to express whether the projects received grant funding
projectSavingsToInvestmentRatio	DatatypeProperty	Estimated Savings To Investment Ratio (SIR) of the renovation project
projectSimplePaybackTime	DatatypeProperty	Estimated Simple Payback Time (SPB) of the renovation project
projectStartDate	DatatypeProperty	Date on which the project investment started
projectUsesIncentives	DatatypeProperty	Yes or no data field to express whether the projects benefitted or will benefit from incentive schemes

VIII.4. bigg4kpi

concepts	type	comment
http://bigg-project.eu/ld/bigg4kpi#	http://www.w3.org/2002/07/owl	Ontology
BuildingKeyPerformanceIndicator	Class	A subclass of key performance indicator related to buildings
KeyPerformanceIndicator	Class	An specialization of a s4city Key Performance Indicator
EEMKeyPerformanceIndicator	Class	A subclass of key performance indicator related to energy efficiency measures
KPICalculationItem	Class	KPI calculation item
assessesAggregatedKPI	ObjectProperty	Relation between an analytical group and an aggregated KPI assessment
AnalyticalGroup	Class	Analytical group
AggregatedKPIAssessment	Class	Aggregated KPI assessment
assessesSingleKPI	ObjectProperty	Relation between a KPI calculation item and its single KPI assessment
SingleKPIAssessment	Class	Single KPI assesment
groupsForAnalytics	ObjectProperty	Relation between a KPI calculation item and its analytical group
hasAggregatedKPIPoint	ObjectProperty	Relation between an aggregated KPI assessment with its aggregated KPI assessment point
AggregatedKPIAssessmentPoint	Class	Aggregated KPI assessment point
hasAggregationFunction	ObjectProperty	Relation betweeen an aggregated KPI assessment and its aggregation function
AggregationFunction	Class	A function that aggregates KPI values
hasAnalyticalModel	ObjectProperty	Relation between a KPI calculation item and its analytical model
AnalyticalModel	Class	Analytical model
hasKPI	ObjectProperty	Relation between a KPI calculation item and its KPI
hasKPIType	ObjectProperty	Links a KPI with its type given as a skos concept
hasModelStorageInfrastructure	ObjectProperty	Relation between an analytical model and its model storage infrastructure
ModelStorageInfrastructure	Class	Model Storage Infrastructure
hasModelType	ObjectProperty	Links an analytical model with its type given as a skos concept
hasSingleKPIPoint	ObjectProperty	Relation between a single KPI assessment and its KPI assessment point
SingleKPIAssessmentPoint	Class	Single KPI assesment point
isEstimatedByModel	ObjectProperty	Relation between a KPI assessment and its analytical model
KeyPerformanceIndicatorAssessment	Class	An specialization of s4city key performance indicator assessment
modelBaselineYear	DatatypeProperty	The baseline year of an analytical model
modelLocation	DatatypeProperty	The location of an analytical model
modelName	DatatypeProperty	The name of an analytical model
modelTrainedDate	DatatypeProperty	The training date of an analytical model
TimeSeriesPoint	Class	A class that represent a single point of a time series set of data
TimeSeriesList	Class	A class that represent the whole set of time series points as a list

VIII.5. bigg4co2

concepts	type	comment
http://bigg-project.eu/ld/bigg4co2#	http://www.w3.org/2002/07/owl	Ontology
appliesTo	ObjectProperty	The commodity to which a CO2 equivalent emission factor applies to.
Co2EquivalentEmissionsFactor	Class	The CO2 equivalent emissions factors are rates that quantifies a public utility environemental impact. It can be based on time, thresholds or consumption; however, combined CO2 equivalent emissions may also exist that mix more than one of these types.
forAbsoluteTimeAtDay	ObjectProperty	The time interval in each day for which a CO2 equivalent emissions factor is applied (e.g., 8:00 to 10:00).
TimeBasedCo2EquilvalentEmissionsFactor	Class	A time-based tariff is a tariff that is based on time.
forWeekDay	ObjectProperty	The day of the week for which a CO2 equivalent emissions factor is applied (e.g., each Saturday and Sunday).
hasDuration	ObjectProperty	The duration of a CO2 equivalent emissions factor.
hasPeriod	ObjectProperty	The period of a CO2 equivalent emissions factor.
forDayInMonth	DatatypeProperty	The day of the month for which a CO2 equivalent emissions factor is applied (e.g., each 15).
forVolumeConsumption	DatatypeProperty	The volume consumption related to a CO2 equivalent emissions factor (e.g., after consumption of 1 ton).
ConsumptionBasedCo2EquivalentEmissionsFactor	Class	A consumption-based CO2 equivalent emissions factor is a factor that is based on consumption.
forVolumeFlow	DatatypeProperty	The volume flow related to a CO2 equivalent emissions factor.
ThresholdBasedCo2EquivalentEmissionsFactor	Class	A threshold-based CO2 equivalent emissions factor is a factor that is based on a threshold.
hasStartTimestamp	DatatypeProperty	The start date and time of a CO2 equivalent emissions factor.
Co2EquivalentEmissions	Class	Equivalent emissions of CO2.
GridDeliveredElectricity	Class	Electricity delivered from the grid as a source for CO2 emissions.
LPG	Class	Liquified Petroleum Gas as a source for CO2 emissions.
NaturalGas	Class	Natural gas as a source for CO2 emissions.
Propane	Class	Propane as a source for CO2 emissions.