BIM4EEB

BIM based fast toolkit for Efficient rEnovation in Buildings

D3.6 Integrated Linked Data Modelling and Sharing Framework

This project has received funding from European Union's H2020 research and innovation programme under grant agreement N. 820660

Programmes	H2020
Call for Proposal	LC-EEB-02-2018 Building information modelling adapted to efficient renovation
Project Title	BIM based fast toolkit for Efficient rEnovation in Buildings
Acronym	BIM4EEB
Project Grant Agreement	820660

D3.6 Integrated Linked Data Modelling and Sharing Framework

Work Package	WP3
Lead Partner	TU Dresden
Contributing Partner(s)	PoliMi, VTT, UCC, Suite5, VisuaLynk
Dissemination Level	Public
Type	Report
Due date	31/12/2019
Updated version 3.0 submission date	07/02/2022
Version	3.0

DOCUMENT HISTORY

Version	Date	Comments	Main Authors
0.1	02.09.2019	ToC Created	TUD
0.2	17.09.2019	Revised ToC	TUD
0.3	15.10.2019	Data specification requirements for use cases	All partners involved in T3.6
0.4	30.11.2019	Revised ToC considering feedback received from EC-Reviewer	VisuaLynk, TUD, Polimi
0.5	13.12.2019	Draft Status after ToC revision	All partners involved in T3.6
0.6	23.12.2019	Sections 4.3.6 and 6.1 substantially revised	VisuaLynk
0.7	21.01.2020	The methodological part (chapter 3) was substantially revised. Chapter 4 re-structured and updated	TUD
0.8	27.01.2020	The re-structured version presented for final revision	TUD
1.0 FINAL	31.01.2020	Comments from (Internal Reviewers) integrated and submitted to EC	TUD (Polimi, VisuaLynk)
1.1	15.6.2020	Appendix added (validation achievements)	TUD (Suite5, VTT, VisuaLynk)
1.2	23.7.2020	Final corrections to appendix	TUD
1.3 FINAL	27.7.2020	As submitted to co-ordinator and resubmitted EC	TUD
1.4	03.05.2021	The framework for the linked data modeling is updated. Alignments between the BIM4EEB and External Ontologies	P. Valluru (TUD), S. Törma (VisuaLynk), J. Karlapudi (TUD)
1.5	12.05.2021	Changes have been included by considering reviewer comments, submitted for internal review	P. Valluru (TUD)
1.6	24.05.2021	Internal review	D. Madeddu (OneTeam), S. Törma (VisuaLynk)
1.7	25.05.2021	Contribution to BIMcpd, BIM4Occupant tool sections	P. Valluru (TUD), A. Hryshchenko (UCC), K. Tsatsakis (Suite 5)
1.8	27.05.2021	BIM4EEB tools and Usage of ontologies in tools	Prathap Valluru (TUD)
1.9	31.05.2021	Ontology Evaluation based on competency questions. The diagrams and bibliography updated	P. Valluru (TUD), S. Törma (VisuaLynk), J. Karlapudi (TUD)
2.0	14.06.2021	Review of PoliMi's team including PC. Plagiarism check. The final version with all required updates	P. Valluru (TUD)
2.1	14.06.2021	References to other activities and the state of the art	S. Törma (VisuaLynk)
2.2/2.3/2.4	25-28.06.21	Updates and Revision by WPL	K.Menzel, P.Valluru (TUD)
2.5	30.06.2021	Final revisions by WPL and PC approval.	K.Menzel, P.Valluru (TUD)
2.6	06.12.2021	Revised the document by considering the monitor comments	P.Valluru (TUD)
2.6	11.01.2022	Internal review	K.Tsatsakis (Suite5), S.

			Törma (VisuaLynk), K.Menzel (TUD)
2.7	02.02.2022	Revised document by considering internal review comments WP Leader's approval	P.Valluru (TUD), S. Törma (VisuaLynk)
3.0 FINAL	03.02.2022	Final Version, finalized after final and formal review and plagiarism check	B.Daniotti, M.Signorini (Polimi), G.Canzi (FPM/Polimi)

Statement of originality:
This deliverable contains original unpublished work except where indicated otherwise. Acknowledgment of previously published material and of the work of others has been made through appropriate citation, quotation, or both.

BIM4EEB action has received funding from the European Union under grant agreement number 820660.

EXECUTIVE SUMMARY

This document presents the Linked Data and ontology framework defined in BIM4EEB for semantic interoperability between the tools developed for the management of renovation projects.

In the context of the AEC & FM sector, the linked data approach has the potential to become the foundation for a strong, comprehensive knowledge base documenting different types of built artifacts. From a BIM4EEB perspective, it is desirable to compile relevant data about the as-built status of a building before the renovation will start. However, one cannot expect that as-built data will be available (or was compiled) over a long operational period in one single, standardized data exchange format or even information model. Thus, there is a need to develop a framework that allows stakeholders involved in a renovation project to efficiently compile, maintain and add data about (i) building elements, (ii) building services systems, (iii) the tenants, operators, and owners of the building, and (iv) the current and predicted performance of the building from the various data sources available.

Over the past three decades, a comprehensive body of knowledge in open Building Information Modelling has been developed, maintained, and documented by Building Smart International. One major component is the IFC meta-data model. This model is implemented in different modeling languages, such as EXPRESS, XML, or OWL. However, published versions of this information model do not fully exploit the potential for knowledge management. The expressivity and complexity of available model implementations are high, and consequently, the performance of model analysis activities decreases and makes the efficient usage of these models especially for knowledge management tasks (e.g. reasoning) less comfortable. Apart from the IFC model and its complexity, there are questions of the representation of construction plans, linking them in a proper way to other entities, representation of units, observations, time, and so on.

Therefore, we propose in this deliverable the usage of Modular Ontologies and linked data framework for the BIM4EEB project. This deliverable starts with the state of art analysis for ontology modularization. Later, we introduce a framework for linked data modeling and sharing for the BIM4EEB Project. Based on the findings in previous deliverables of WP3 we present a set of Ontologies that were developed by the BIM4EEB community to support specific activities typical for the renovation of residential buildings. We illustrate, how these ontologies integrate “third party ontologies” as a whole or in part to maximize the re-use of existing models and concepts.

Additionally, we introduce an evaluation methodology for ontologies by using competency questions. In this step, we present a set of competency questions that specify the information requirements of Use Case Scenarios earlier specified in WP2 of BIM4EEB. This is complemented by a presentation of alignment rules which can be used to integrate relevant concepts of all ontologies, thus contributing to the formation of the BIM4EEB Linked Data Modelling and Sharing Framework.

The ontologies developed in WP3 will be supported in other BIM4EEB-work packages, e.g. in WP4, WP6, and WP7 for the semantic knowledge. As a part of ontology evaluation, example data is created. Ontologies are queried using SPARQL queries.

PUBLISHING SUMMARY

This Deliverable presents the BIM4EEB approach for the development of a Linked Data Modelling and Sharing Framework to be used in renovation projects for residential buildings aiming to improve the present renovation scenarios, example workflows, energy efficiency, ..etc. One major “road block” to organize the planning, design, and execution of renovation activities holistically and comprehensively is the lack of efficient data and information sharing. The work in Task 3.6 in WP3 of the BIM4EEB-project aims this deficit.

The proposed approach using the concept of Modular Ontologies will support the integration of available knowledge repositories, describing the “pre-renovation” status of buildings. Thus, decisions about what and how to renovate can be made faster and in an “informed way”. Secondly, it is known from the literature that user behavior impacts the energy consumption of buildings. Therefore, a substantial part of the ontologies developed in WP3 focuses on “occupant behavior modeling” and “user comfort modeling”. A third pillar in the development of the BIM4EEB modular ontology is the knowledge management for “workflow and process management”.

The achievement presented in this deliverable is a Linked Data Modelling and Sharing Framework based on a “Modularly Networked Set of Ontologies” (Digital Construction Ontologies) consisting of ontologies developed by BIM4EEB complemented by ontologies developed by “Third parties” and already well established in the W3-community.

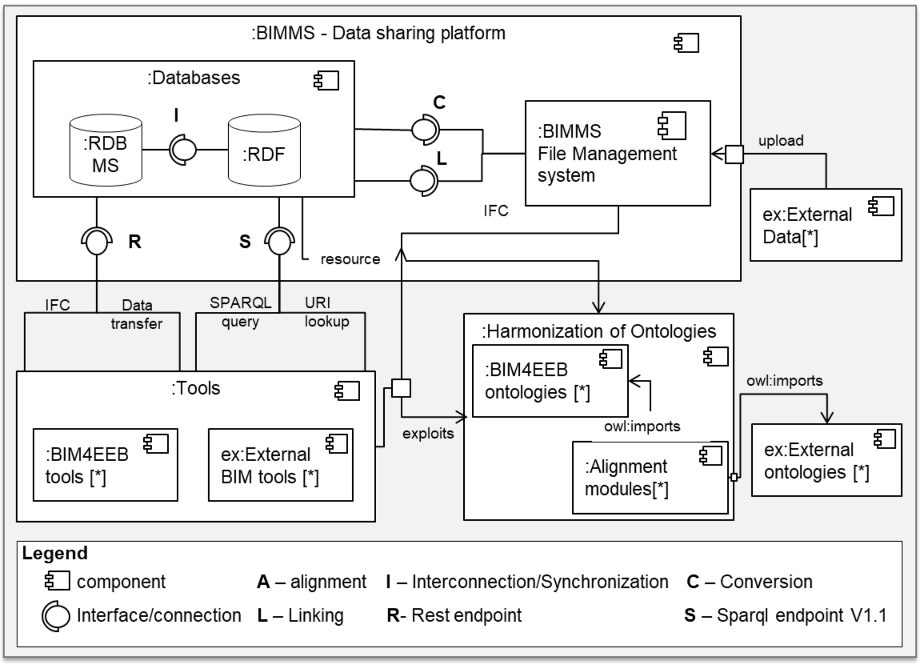

Figure 1: The BIM4EEB-Linked Data Modelling and Sharing Framework (notation: UML)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

In recent years attention has increasingly focused on BIM as a means of developing task-based product models, solving interoperability problems, which has long been a challenge in information-sharing technologies between stakeholders of the AEC/FM domain. Substantial progress has been made over the past three decades, including

(1) a paradigm shift from geometric modeling to object-oriented modeling. This puts the “engineering artifacts” (or parts of them) in the focus of the work, rather than their geometrical representations.
(2) The introduction of “open” meta-data models which are standardized on an international level (IFC: Industry Foundation Classes). This allows the AEC/FM community to maintain and exchange product and process models in “vendor-neutral” formats [ISO16739,2013].
(3) The development of complementing standards for different purposes, such as:
    (3.1) Information Delivery Manual (IDM): Defines & documents processes and data requirements.
    (3.2) Model View Definitions (MVD): This is a specification for data model exchange.
    (3.3) BIM Collaboration Format (BCF): Model-based, neural communication protocol,
    (3.4) buildingSMART Data Dictionary (bSDD): Standard library with general definitions of BIM objects and their attributes.

The above set of methods and tools has been developed under the auspices of Building Smart International, a non-profit organization acting on a global level. In the context of this deliverable, the achievements of Building Smart can be interpreted as a “consolidated, formalized set of information requirements specifications” summarising the findings of a large AEC/FM community.

Semantic web technologies are one of the emerging solutions to exchange information between tools and systems. In one sentence, the term Semantic Web technology can be explained as “a way of linking data between systems or entities that allows for rich, self-describing interrelations of data available across multiple stakeholders on the web.” Linked data is capable of carrying the information in the form of triples (subject, predicate, object). It was the focus of BIM4EEB’s task 3.6 and thus guides the work in this Deliverable D3.6.

Current work in the area of Linked Data in AEC/FM is driven by the Linked Building Data (LBD) community group at the World Wide Web Consortium (W3C).

Work on Semantic Web Technologies is motivated to overcome the identified shortcomings of the current implementations or specifications of the above-mentioned IFC-Meta Model. Some authors [Bonduel, 2018], [Beetz, 2009] identified specific drawbacks in the IFC meta-data schema in terms of providing efficient data management and modeling, such as:

- The EXPRESS and XSD languages lack methods for defining formal semantics, making it difficult to apply generic reasoning and querying methods on IFC building models.
- Developers can propose extensions for the IFC schema to buildingSMART, but the technology does not allow an extension of the IFC schema on the fly in a user-friendly way.
- Fine-grained linking of building information stored in an IFC file to related data on the web (e.g. regulations of local authorities, geographic information, general knowledge, etc.) is not possible.
- Lack of formal rigidness
- Limited reuse and interoperability
- Lack of built-in distribution

WP 3 of BIM4EEB addresses the efficiency gaps of current data-sharing technologies in BIM. So far we analyzed existing ontologies relevant to the renovation domain and developed proposals for refinements and extensions of existing ontologies. Furthermore, we investigated what inter-model and inter-ontology relationships are required to integrate concepts (relevant to the objectives of BIM4EEB, such as comfort, occupancy, energy performance, delivery phase processes, materials, equipment) which are absent or underdeveloped in current ontologies. To achieve the final goal, i.e. a flexible and easily adaptable framework allowing to share relevant data during the time of its publication or on-demand task 3.6 focused on the harmonization of relevant ontologies to present a Data Modelling and Sharing Framework using Linked Data Principles. The results are presented in this Deliverable.

So far, the work in WP 3 has progressed as shown in Figure 2. Task 3.1 proposed the general framework for linked data, which is identifying the vocabulary requirements in the renovation domain, analyzing the existing ontologies, and single out the missing data in the current renovation domain ontologies. In addition to that, it also discussed current practices for ontology development.

In task 3.2 the first domain-specific ontologies were developed for the BIM4EEB renovation domain with an emphasis on modeling occupant behavior, building services systems, and user comfort. Task 3.3 elaborated the level of details for a possible building renovation product-process ontology. In a later step, the development of the renovation workflow ontology progressed in task 3.4. Finally, task 3.5 discusses the expansion of measurement and verification protocols for local energy performance models. Figure 2 represents the steps graphically and illustrates the role of this deliverable D 3.6 after the completion of Task 3.5.

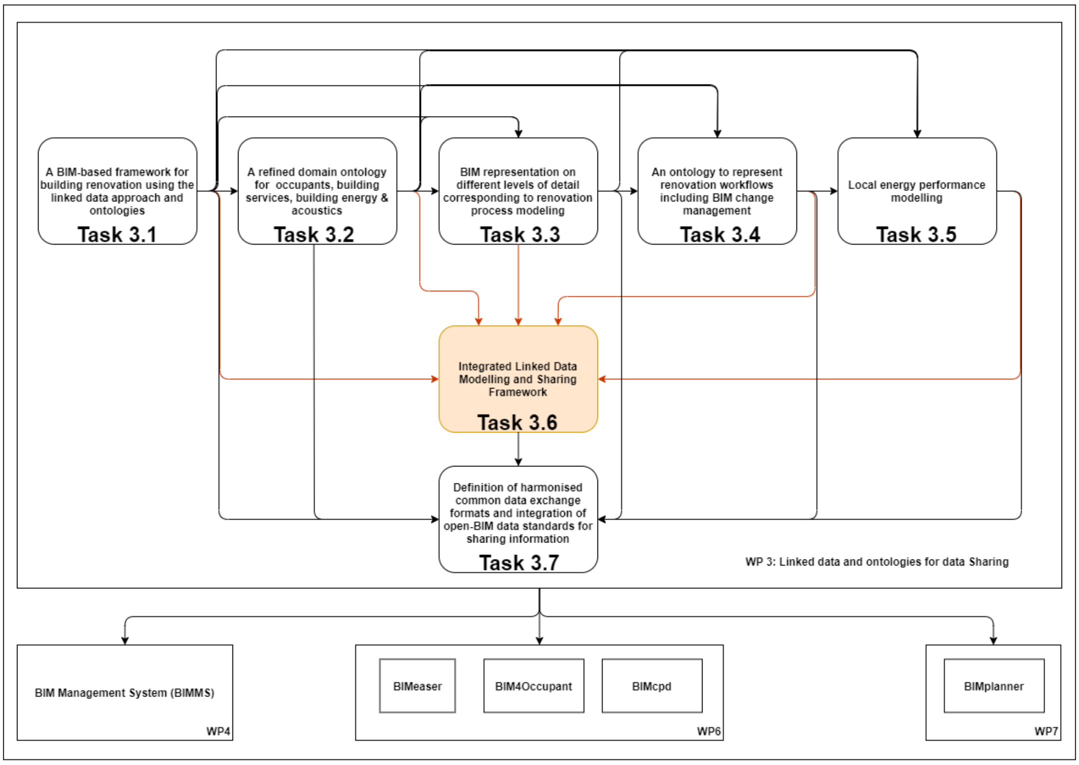

Figure 2: Workflow diagram of WP 3: linked data and ontologies for interoperability

The current task 3.6 explains the harmonization of ontologies by considering developments in the previous tasks and also ontology evaluation is carried out. It is complemented by efforts in task 3.7, in which existing regulatory frameworks and recent research efforts are analyzed. Task 3.7 will also highlight possible features for a new data exchange standard at the European level (CEN TC442).

The ontologies developed in WP3 will be exploited later in the project at the work performed in WP4, WP6, WP7. The BIMMS – developed in WP4 – supports ontologies as a resource and also supports linked data features, such as named graphs. The tools developed in WP6, e.g. BIMeaser (BIM early-stage energy scenario tool), BIM4Occupant, and BIMcpd (BIM Constraint Checking, Performance Analysis, and Data Management) inherit the ontology model defined in order to address linkages among data. The BIMPlanner tool developed as part of WP7 will use ontologies to support planning and site progress operations.

1.1 Work results and progress

Digital Construction Ontologies (DiCon) acts as an enabler of semantic interoperability between systems in the construction and renovation domain. They define the basic terms related to built assets, building design, and construction planning, including the agents, resources, and Spatio-temporal aspects, materials, lifecycle data, occupancy, acoustic, energy aspects. The definitions rely heavily on standards (ISO 19650 BIM/IM, ISO 16739-1 IFC, ISO/IEC 21838-2 BFO, and ISO 21597-1 ICDD) and established ontologies (such as OWL-Time, PROV-O, SSN/SOSA, etc). The correspondences of Digital Construction Ontologies with other ontologies are provided explicitly in separate alignment modules.

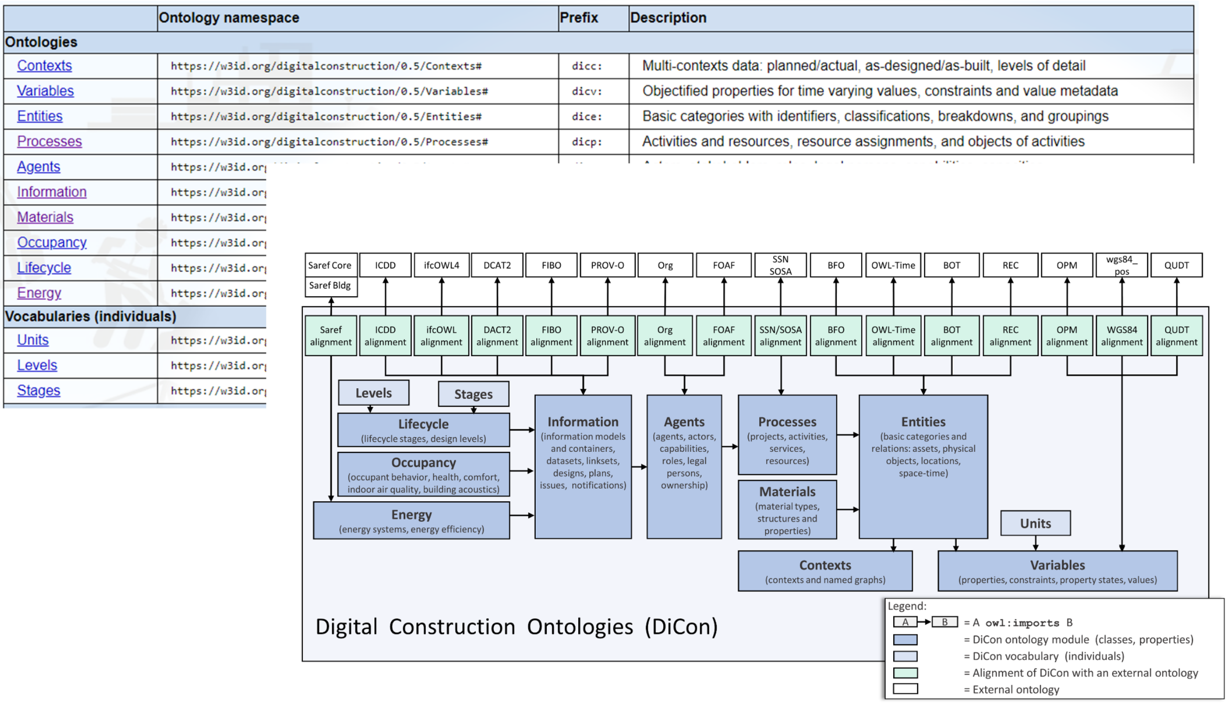

Figure 3: an overview of the ontologies and their publication on GitHub page

The specific objective of Deliverable D3.6 is to cover the representation of digitalized construction processes. Digital technologies are increasingly used in all stages of the construction lifecycle, from scanning, building information modeling, and procurement to supply-chain management and construction management. Mobile devices, sensors, and imaging solutions provide a flood of instantaneous data from project execution. The variety, volume, and velocity of digital information are growing, which calls for automatization of information sharing between tasks and systems. Digital Construction Ontologies are designed to provide a common terminology needed in machine-to-machine information flows.

Digitalized construction projects can employ a wide range of task-specific point solutions that can significantly enhance the productivity and quality of individual tasks. However, the improvement of overall construction productivity depends crucially on what happens between the tasks: how the information produced in one task can be utilized in other tasks. By improving the semantic interoperability, the Digital Construction Ontologies are becoming an enabler for smoother information sharing and eventually higher productivity and quality of construction and renovation projects. Similarly, ontologies are used for data sharing in improving the early-stage design process.

2 Ontology Harmonization – A state of the art analysis

The concept of the Semantic Web and the Linked Data paradigm is the technical advancements that enable easier access to the data and improved data, information, and knowledge sharing by adding contextual information to data and information. These concepts also enable improved interoperability between different applications or IT tools.

The goal of the BIM4EEB project is to apply the concept of Semantic Web and the Linked Data paradigm to building renovation to achieve more efficient data management and improved data interoperability.

The acquisition of data from different sources is a usual process in the construction industry and thus also in building renovation. It is also well known that numerous domains, tools, experts, data sources, etc. are involved in buildings' renovation. Therefore, we aim to develop a set (or suite) of modular but integrated ontologies which support both the particular domains and through their integration efficient and seamless sharing of information and knowledge. Individual, modular ontologies are dedicated to supporting the specific knowledge representation of a selected domain. They enable the involvement of different domain experts and experienced professionals in the knowledge representations.

Additionally, it is also important to have appropriate instruments for ontology development. In the BIM4EEB project Competency Questions (CQ) are used to define information requirements of domain ontologies and to evaluate the functionality of developed ontologies.

Finally, it is recommended to use the already existing ontologies, which were developed by experts for general purposes or the usage in related domains. This can reduce own development efforts but also reduces the complexity of information modeling [Smith,2008].

Barry Smith and Mathias Brochhausen [Smith,2008] state that the following objectives must be met to make the developed or existing modular ontologies useful:

- First, it is essential to align/match existing ontologies;
- Secondly, it is necessary to find ways to evaluate ontologies transparently.

These two aspects are discussed in the subsequent sections.

2.1 Ontology Alignment

Ontologies' matching is the process that generates an *alignment* A^{l} for a pair of ontologies O and O^{l} [Euzenat, 2011]. To match some things means, "to bring into line". For given two ontologies aligning one ontology with another means that in the first ontology one tries to find a corresponding entity with the same meaning for each entity (concept, relation, or instance) in the second ontology [Said Rabah, 2013].

2.1.1 Ontology alignment techniques

There exist numerous techniques for ontology alignment. First of all, one can distinguish on what level the alignment shall be executed, such as:

- Terminological Level (i.e. to identify similar / different names).
- Descriptive Level (i.e. to identify different sets of properties)
- Structural Level (i.e. to identify different constructs used to represent an element)
- Semantic Level (i.e. elements describe different sets of real-world entities).

In Figure 4 the above-mentioned levels are represented in “dotted” rectangles. We also call this the “2nd level” (see Table 1).

For the categorization of alignment techniques, it is also important to distinguish on a higher level between

(1) The granularity on which the alignment shall be executed (single element (A-Box) or structure level (T-Box)).

(2) The Origin or Kind of Input, i.e. Content-based for the single (structural) element or Context-based for the structural element and its relationships to other structural elements.

This is also called the “First Level” in Table 1.

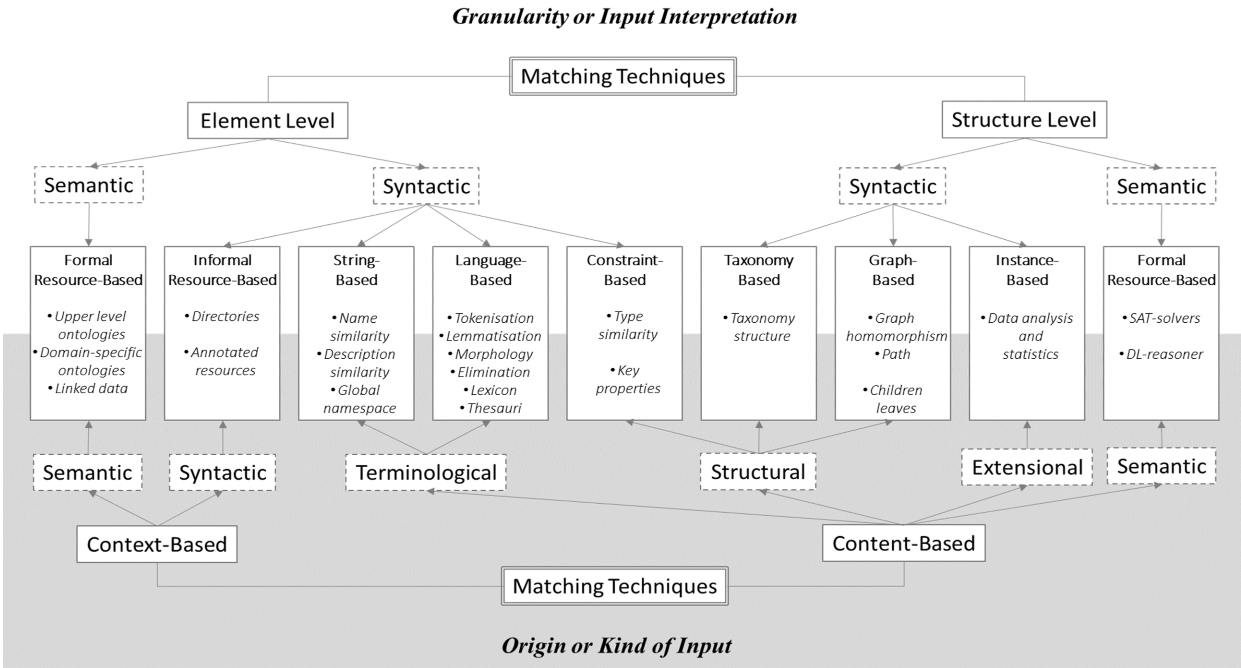

Figure 4: Matching techniques classification [Otero-Cerdeira,2015]

The below tables explain the characteristics of the alignment categories (1st, and 2nd Level) and the specific alignment techniques available (Basic Level in Table 1 and center of Figure 4).

Table 1: First and second level categories of matching techniques [Otero-Cerdeira,2015]

Name	Level	Description
FIRST LEVEL		
Granu-larity	Element level	Objects are considered in isolation in the ontologies while ignoring that they form part of the ontology structure.
	Structure level	To investigate how the entities fit in the structure of the ontology.
Kind of input	Context-based	To consider the alignment of outer data, originating from relations between ontologies or other outer assets (context).
	Content based	To focus on inside data, originating from the ontologies to be aligned.

Table 1 (cont.): First and second level categories of matching techniques [Otero-Cerdeira,2015]

Name	Level	Description
SECOND LEVEL		
Granu-larity & Kind of input	Syntactic	Information elucidation limited to directions expressed in their relating strategies
	Semantic	to utilize conventional semantics to translate information & legitimize the outcomes
Kind of input	Terminological	to consider input as strings
	Structural	Strategies depend on the structure of the entities (classes, individuals, relations).
	Extensional	to compute correspondence by analyzing the instances of the classes (extension)
	Semantic	To use a reasoned to deduce correspondence (i.e. to provide linguistic feedback).
BASIC LEVEL (Concrete matching techniques)		
Basic	Formal Resource-based	Structured resources are used to support matching processes (e.g. upper-level ontology, domain-specific ontology, or aligned ontologies (reuse).
	Informal Resource-based	Like in the previous category □ to take advantage of external resources, but in this case on an informal basis. To explore the relationship between ontologies and such informal tools to deduce relationships between ontologies.
	String-based	to compare between the strings describing entities' names and definitions
	Language-based	Based on Natural Language Processing. Names are regarded as words in a natural language (rather than strings).
	Constraint-based	To consider parameters relating to the internal structure of objects (e.g. domain and the range of properties, types of attributes).
	Graph-based	to treat ontologies as labeled graphs, or even trees, and treat the alignment problem of ontologies as a graph homomorphism
	Taxonomy-based	these methods can be viewed as a specific instance of the past ones which just think about the specialization connection.
	Instance-based	to exploit the augmentation of the classes in the ontologies, i.e. in the event that individuals are indistinguishable, the classes might be likewise not comparable.
	Model-based	to exploit the semantic understanding connected to the input ontologies.

2.1.2 Selected ontology alignment axioms

The table below provides an overview of axioms which can be used to specify Ontology Alignments.

Table 2: Ontology alignment axioms

	No	Language Feature	Functional Syntax	RDF Syntax	Description
Class Expression Axioms	1	SubClass	SubClassof (C1 C2)	C1 rdfs:subClassOf C2.	Used to state that all the instances of one class are instances of another.
	2	Equivalent classes	EquivalentClasses (C1 ……. Cn)	Cj owl:equivalentClass Cj+1. J=1…..n-1	Used to state that two classes are equivalent amounts the same as stating that both Cj is subclass of Cj+1 and Cj+1 is a subclass of Cj
Object Property Axioms	3	Subproperty	SubObjectPropertyOf (P1 P2)	P1 rdfs:subPropertyOf P2	Used to state that all resources related by one property are also related by another.
	4	Equivalent Properties	EquivalentObjectProperties (P1 … Pn)	Pj owl:equivalentProperty Pj+1. j=1…n-1	Used to states that all of the object property expressions OPEi, 1 ≤ i ≤ n, are semantically equivalent to each other. The axiom EquivalentObjectProperties(OPE1 OPE2) is equivalent to the following two axioms: SubObjectPropertyOf(OPE1 OPE2) SubObjectPropertyOf(OPE2 OPE1)
Data Property Axioms	5	Subproperty	SubObjectDataPropertyOf (P1 P2)	P1 rdfs:subPropertyOf P2	Used to state that all resources related by one property are also related by another.
	6	Equivalent Properties	EquivalentDataProperties (P1 … Pn)	Pj owl:equivalentProperty Pj+1. j=1…n-1	Used to states that all of the object property expressions OPEi, 1 ≤ i ≤ n, are semantically equivalent to each other. The axiom EquivalentObjectProperties(OPE1 OPE2) is equivalent to the following two axioms: SubObjectPropertyOf(OPE1 OPE2) SubObjectPropertyOf(OPE2 OPE1)
Assertions	7	individual equality	SameIndividual (a1 … an)	aj owl:sameAs aj+1. j=1…n-1	Used to state that two names refer to (denote) the same individual.

2.2 Ontology Evaluation

Evaluation of ontologies is the task of measuring an ontology's quality. The evaluation can be executed from different views [Vrandecic,2009]. As one can see in Figure 5 the Ontology Evaluation Methods (OEM) is in the center of the evaluation framework. The OEM can be grouped into Ontology Evaluation Criteria (OEC), each of them being part of Ontology Evaluation Perspectives (OEP).

Complementing the grouping of OEM in OEC one can use different Ontology Evaluation Approaches (OEAs), again, each of them being applicable to different Ontology Evaluation Aspects (OEAs). In the following sections, we will provide a detailed explanation of these different views on Ontology Evaluation.

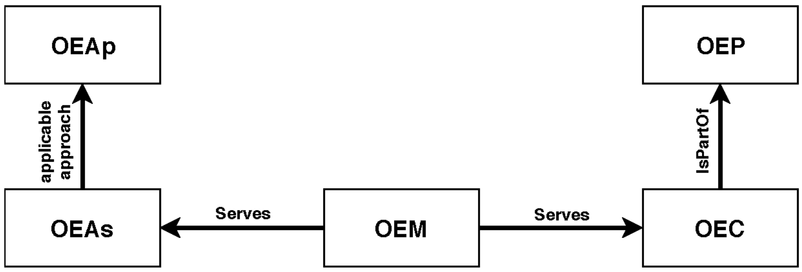

Figure 5: Five views of ontology evaluation

2.2.1 OEM: Ontology Evaluation Methods

In the literature, one can find more than twenty evaluation methods for ontologies. These methods are used to evaluate one specific view on ontologies. Before we explain different groupings of these methods we provide an overview of these methods in a tabular format (see Table 3). The methods are already grouped according to OEAs (see Table 3).

Table 3: Overview of OEM (grouped as per OEAs)
(see also part 2 overleaf)

Nr.	Name	Description
OEAs: VOCABULARY		
1	Check used protocols	All URIs are tested for well-formed URIs in an ontology.
2	Check response codes	Create a HEAD call on them for all HTTP URIs. Names with the same namespace slash would return the same answer codes, otherwise, an error would be indicated. (Applicable to the ontologies published on the web).
3	Look up names	Make a GET call against the namespace for every name that has a hash namespace. Make a GET call against the name for each name that has a slash namespace. The type of content should be set properly. If any, address redirects. If the asset returned is an ontology, test if the name is defined by the ontology. If so, N is a name that corresponds to the linked data. If not, the name may be incorrect
4	Check naming conventions	By comparing the local part of the URI with the label given to the entity or using lexical tools, a proper name can be verified. Notice that only local names from the same namespace must be used uniformly, not all local names in ontology, i.e. names reused from other ontologies may use different naming conventions.
5	Metrics of ontology reuse	Finding namespaces used in the ontology, name references to unique names, unique URIs to namespaces.
6	Check name declarations	Check for each URI if a URI claim exists. If so, check that the stated type is usage-consistent. This way, punning wrongly implemented can be identified.

Nr.	Name	Description
7	Check literals and data types	A set of data types permitted should be created. It is necessary to avoid all data types beyond those specified in the OWL specifications. There should be a very strong reason to create a custom data form. xsd: integer and xsd: string should be the preferred type of data (because all OWL-conforming methods must be implemented).
8	Check language tags	Verify that the definition of all language tags is correct. Verify if a language tag is used as short as possible (i.e. delete unnecessary data such as restating existing scripts or default regions). Test if the word and script used in the literal is the one used.
9	Check labels and comments	Defines the set of applicable ontology languages. Ensure that the language is labeled for all tags and literal statements. Check that all entities have a relevant label in all languages. Check if there is one in all relevant languages for all entities that need a comment. Review whether the labels and comments obey the ontology style guide.
10	Check for superfluous blank nodes	List all cases of blank nodes in RDF graphs that are structurally required. Verify that it belongs to one of these cases for each blank node. Besides those, there should be no other empty nodes in the RDF map. As potential errors, all empty nodes that are not structurally necessary should be identified.
OEAs: SYNTAX		
11	Validate against an XML schema	In specific circumstances, an ontology can be validated using a generic XML validator.
OEAs: STRUCTURE		
12	Ontology complexity	Defines measures considering the presence of each ontological language function.
13	Search for Anti-Patterns	SPARQL queries can be used to identify potentially problematic trends over the ontology network.
14	OntoClean meta-property check	An ontology can be marked with the meta-properties of OntoClean and then automatically reviewed for breaches of restrictions.
OEAs: SEMANTICS		
15	Ensure a stable class hierarchy	Calculate a normalized class depth measure, i.e. calculate the length of the longest subsumption path on the normalized version of the ontology md(N(O)). Now calculate the stable minimal depth of the ontology mdmin(O). If md(N(O)) not equal to mdmin(O) then the ontology hierarchy is not stable and may collapse.
16	Measure language completeness	Checking language completeness over language fragment.
OEAs: REPRESENTATION		
17	Explicitness of sub-sumption hierarchy	New metric to describe explicitness of the subsumption hierarchy.
18	Explicit terminology ratio	Finding the ratio of classes and class names, the ratio of properties and property names to check normalized and original ontology.
OEAs: CONTEXT		
19	Check competency questions against results	Form the competency questions and write down expected answers then cross-check the results from the ontology query.
20	Check competency questions with constraints	Formalize your ontology O competence question as a SPARQL CONSTRUCT query formulating the result as ontology R in RDF. Merge R with O and possibly an empty ontology with additional limitations C. Search for contradictions in the merged ontology.
21	Unit testing with test ontologies	Unit tests for ontologies will help in the case of dynamic ontologies.
22	Increasing expressivity	A strongly axiomatized version of the ontology, C will follow an ontology O. O∪C merged ontology must be accurate, otherwise, the contradictions lead to errors in O.
23	Inconsistency checks with rules	Translate the ontology to be analyzed into a logic system and probably restrict ontologies. It's not necessary to complete this translation. Formalize external restrictions as guidelines or limits of integrity.

2.2.2 OEP: Ontology Evaluation Perspective:

Ontologies are regarded as reference models, Two essential perspectives must be taken into account to ensure their evaluation [Hlomani,2014], [Raad,2015], they are (a) Quality, and (b) Correctness.

2.2.3 OEC: Ontology Evaluation Criteria

The ontology evaluation perspective quality and correctness deal with several parameters [Vrandecic,2009], [Hlomani,2014], [Raad,2015]. Those parameters are considered under the ontology evaluation criteria [Vrandecic,2009]. The OEC parameters are listed as below:

a) **Accuracy:** is a criterion that states whether the ontology axioms comply with the domain knowledge of the stakeholders.

b) **Adaptability:** Measures the degree to which ontology predicates its uses.

c) **Completeness:** Measures whether this ontology sufficiently covers the domain of interest or not. An ontology can provide the theoretical framework for a variety of planned tasks.

d) **Computational efficiency:** Measures the ability of the tools used to work with ontology, especially the speed required by the reasoners to perform the tasks required.

e) **Conciseness:** The criteria that specify that ontology includes irrelevant domain elements to be guarded.

f) **Consistency:** Describes the ontology contains no inconsistencies or permits them.

g) **Clarity:** Measures how effectively ontology communicates the defined terms' intended meaning. Definitions should be objective and context-neutral.

2.2.4 OEAp: Ontology Evaluation Aspects

Evaluation aspect	Description
Vocabulary	The vocabulary of an ontology is the collection of all names in the ontology.
Syntax	
Structure	An RDF graph can be used to define a web ontology. The ontology structure is this graph. The structure can vary highly, even if the same ontology is represented semantically.
Semantics	A consistency ontology is represented to a nonempty set of possible models, usually infinite. The ontology's semantics are common features of all these models.
Representation	This element captures the structure-semantics relationship. Representative aspects are typically evaluated by comparing metrics measured on the RDF graph with features of possible models as defined by the ontology.
Context	This dimension concerns the characteristics of ontology relative to other objects in its context, which can be, for example, a data source defined by ontology, a specific representation of the information within the ontology, or formalized ontology criteria in the form of questions of competency or additional semantics.

2.2.5 OEAs: Ontology Evaluation Approach

Vrandecic [Vrandecic,2009] distinguishes into different ontology evaluation approaches, such as.

Gold standard:

The gold standard offers the greatest path to ontology evaluation by providing the verification ability from the vocabulary to the syntax, semantics, and representations verification. The definition of these verification parameters is represented in the table below.

This approach aims to identify the similarities between the ontologies at the lexical and conceptual levels.

This approach typically follows the comparison of the target ontology with the “gold-standard” which is suitably designed for the domain of discourse [Brank,2005] [Ensan,2013]. This means that a “gold-standard” is considered to be a well-constructed ontology and to be used as a reference for the evaluation of developed ontologies. However, this approach has some constraints in terms of the evaluation of gold standard itself [Vrandecic,2009].

Application based:

The application-based ontology evaluation approach majorly concentrates on the feasibility of ontology in usage and adaptability in the applications.

This approach also focuses on the verification of vocabulary correctness, semantic definitions, and the representations within the ontology. The approach is also be used based on different tasks involved in the represented use-case scenarios.

The application-based ontology has two major limitations or drawbacks. It is always been an arguable topic like what is applicable for one application context is may not apply for the other. This makes harder the generalization of the evaluation process to all ontologies. The other issue with the suitability of a single harmonized ontology. This approach is highly suitable for the small modular ontologies and difficult to manage more number of small ontologies in an automated setting [Vrandecic,2009].

Data driven:

As the name suggests, this evaluation approach typically involves the comparison of the ontologies with the needed or specific domain data. In other words, this approach determines if an ontology refers to a particular topic of interest.

This approach also analyses the ontology efficiency in terms of how far it covers the specific domain. However, the major drawback in the data-driven ontology evaluation is that domain knowledge is implicitly considered as constant but in reality the domain knowledge is dynamic.

User based:

This evaluation process is based on the expertise of the users and their experience in ontology understanding and evaluation. The table below clearly represents the comparative usage of the different approaches in the process of ontology evaluation.

The user-based evaluation approach also concentrates on the context (capturing the subjective information about the ontology) of ontology apart from the other evaluation approaches. The limitation in this approach is the identification of perfect users who can efficiently contribute. It is also difficult to establish standards pertaining to the criteria for evaluation.

2.2.6 Summary of ontology evaluation

In Table 4 we summarise the evaluation sections.

For the BIM4EEB-project we recommend using the user-based evaluation approach since Task 3.6 primarily addresses information and knowledge sharing.

Table 4: Views on Ontology Evaluation

OEP	OEC	OEM																						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Correctness	Accuracy			●										●	●				●	●	●	●	●	●
	Completeness			●			●	●		●	●	●	●			●	●	●		●				
	Conciseness					●					●					●		●	●		●			
	Consistency			●	●	●				●			●	●	●		●					●	●	●
Quality	Adaptability						●				●			●		●		●		●		●	●	●
	Clarity	●	●	●	●		●	●	●	●					●				●					
	Computational Efficiency						●	●			●		●				●							
OEAs		Vocabulary										Syntax	Structure			Semantic		Represen-tation		Context				
OEAp	User Based	i										I	i			i		i		i				
	Gold Standard	i										I				i		i						
	Application Based	i														i		i		i				
	Data Driven	i														i		i						

3 Framework for Linked Data in BIM4EEB

The integrated linked data modeling and sharing framework of BIM4EEB is considered from two perspectives in this deliverable. These are a *software framework* and an *ontology framework*.

A software framework is defined as follows, based on a generalization of extensive descriptions in [Balzert 2000], [Bruegge, 2013] and [Stuckenschmidt, 2009]:

> "*In computer programming, a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software. It provides a standard way to build and deploy applications and is a universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate the development of software applications, products and solutions. Software frameworks may include support programs, compilers, code libraries, toolsets, and application programming interfaces (APIs) that bring together all the different components to enable development of a project or system.*"

As a part of linked data modeling and sharing framework we developed specifications that should be adressed in the tools. These specifications are defined in BIM4EEB deliverable D3.1 and mapped with software framework in the table below.

Ontology frameworks, on the other hand, can be characterized as:

> "***Consistent and operational networks of ontologies*** *reflecting the various spheres of* ***enterprise structures and operations****. Consistent means that the ontologies are based on compatible paradigms, have a* ***compatible degree of detail****, and include at least* ***partial sets of alignment*** *relations which allow data interoperability. Operational means that the ontology specifications are available in a single, current ontology formalism for which* ***scalable repositories, reasoning support, APIs,*** *and* ***tools*** *are available*." [Filipowskaja, 2009]

Table 5 shows different criteria that need to be addressed in the overall linked data modelling and sharing framework. These criteria were defined in the BIM4EEB deliverable D3.1 "A BIM-based framework for building renovation using the linked data approach and ontologies – State-of-the-art, use cases, and high-level architectural specifications".

Table 5: Different criteria considered in the BIM4EEB linked data and sharing framework

		Software Framework	**Ontology Framework**
Objec-tives	To enable semantic interoperability between independent systems in the renovation domain that implements the functionalities and comply with the conventions specified in the framework		
Required functions	Each system that provides data to other systems must: P1. Implement the Linked Data Principles (Berners-Lee, 2006) for data sharing P2. Represent shared data as RDF graphs with links to additional data P3. Support the specified interfaces (SPARQL endpoint, URI Lookup) P4. Use the specified set of shared ontologies to define the properties of entities in shared data	X	

	Each system that consumes data from other systems must: C1. Be able to access the data using specified identifiers, query language, and interfaces C2. Be able to parse the received data and query results C3. Use the specified set of shared O. to interpret the properties of objects in shared data		X	
Conventions	**Identifiers**	HTTP URIs, using the HTTPS protocol. The URI should be based on the GUID	X	X
Conventions	**Data formats**	Turtle for ontologies | JSON-LD, TriG and Turtle for data	X	X
Conventions	**Interfaces**	SPARQL Endpoint | URI Lookup | REST API GraphQL – For complex queries on the REST API	X	
Conventions	**Existing O.**	Use of existing ontologies (O)	X	X
Conventions	**Conceptual gaps that need to be covered by additional ontologies**	G1. Main entities interlinked with activities, external systems, and classifications G2. The types of agents and their roles, capabilities, and production rates G3. Information entities: models, plans, renovation measures, indicators, notifications G4. Different contexts of information: planned/actual, as-designed/as-built, LODs G5. Variables & constraints capturing mgmt. knowledge & evolving designs G6. Occupant behavior and profiles for requirements and evaluation of renovation scenarios, including occupant comfort, indoor air quality, and y G7. Building acoustics, a property affected by renovation measures G8. Energy efficiency and energy systems, central aspects of energy renovations G9. Building materials and their layering, which affects the energy efficiency G10. Building lifecycle and levels of detail, as concrete frameworks G11. Activities and resources, resource assignments, and objects of activities		X
Practices	O: modularization and reuse: The overall O is divided in logical modules based on the vertical and horizontal segmentation. External references made explicit in separate alignment modules.			X
Practices	Ontologies definition: OWL2 DL profile			X
Practices	Ontology metadata provided: License CC-BY, preferred prefixes			X
Practices	Drafting of ontologies: CMapTools			X
Practices	Editing of ontologies: Protégé and a text editor			X
Practices	Documenting of ontologies: pyLODE			X
Practices	Publishing of ontologies: w3id.org/digitalconstruction/			X
Practices	Hosting of ontologies: GitHub Pages			X
Practices	Maintenance of O.: Supports continuous evolution of O. with previous versions backed up			X

For the clear denotation of the BIM4EEB framework, the concept called component diagram (UML diagram) is considered. Major components of this framework are (clockwise)

- BIMMS environment,
- external data (images, files, models, sensor data, etc.)
- External Ontologies,
- BIM4EEB Ontologies, and
- BIM4EEB tools.

The BIM Management System (BIMMS[1]) is a collaborative information sharing environment developed in the BIM4EEB project. It is a platform built around a common data environment (CDE) that stores all the data and information gathered through different sources and throughout the whole building lifecycle, acting as a single source of truth (SSOT) [Alessandro,2020]. The BIMMS supports the storage of various kinds of information: models, plans, documents, drawings, etc. Moreover, it can synchronize the data between relational data structures (RDBMS) and Linked Data (RDF graphs) by providing the linkage between them and converting updates or changes between them. The aim of the BIMMS is the effective management of

[1] https://bim4eeb.oneteam.it/BIMMS/Default.aspx

information produced in the renovation processes along with the establishment of efficient communication between the involved tool kits.

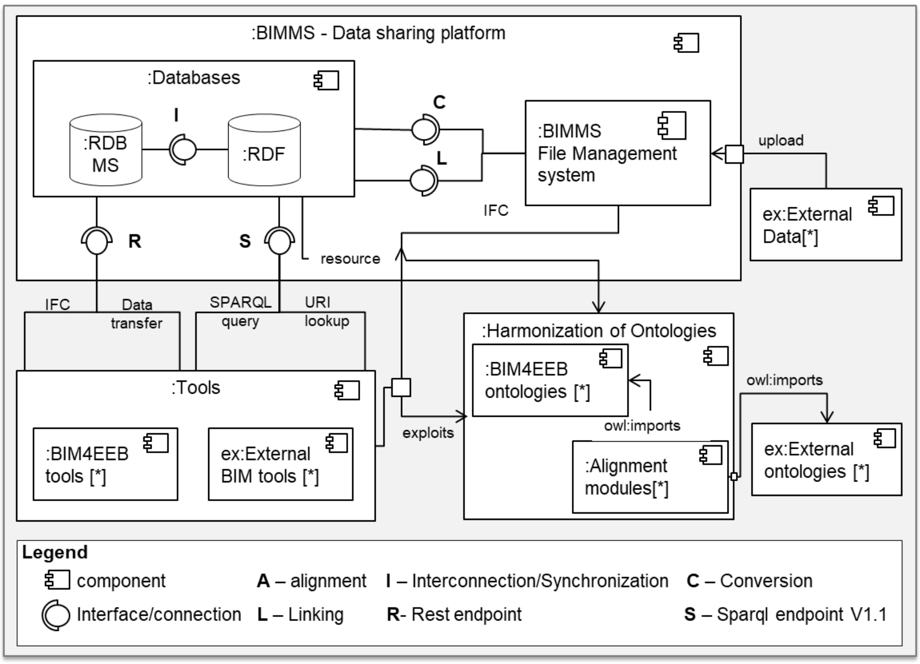

Figure 6: The BIM4EEB-Linked Data Modelling and Sharing Framework

The set of tool have been designed to support important tasks in a renovation process. Data stored in the BIMMS is effectively shared with these tools using the relevant interface connections. In general, there are two main interfaces, such as REST API and SPARQL endpoint. Since majority of current tools are have their own internal data models, the conversion to Linked Data can be managed internally by the tools (in which case it is natural to use the SPARQL endpoint) or by the BIMMS (in which case the use of the REST interface is justified). In the BIM4EEB framework, connections like the REST endpoint, SPARQL endpoint, and URI Lookup are used in the data sharing process between the tools and the BIMMS system. The file formats used to transfer the data is shown in the BIM4EEB D8.1 deliverable.

Hereafter the key components of the ontology framework are further elaborated in relation to the BIM4EEB framework. Moreover, through this discussion we are aiming to deliver a short summary to show that WP3 in BIM4EEB addresses the above criteria.

3.1.1 Enterprise structure and operations

"**Consistent and operational network of ontologies reflecting the various spheres of enterprise structures and operations**". BIM4EEB ontology definition and development was carried out in a relation to the analysis of roles and scenarios in WP2. The main workflow for this analysis process is dependent on different renovation scenarios and their sequences along with the involved stakeholders or actors. This workflow is presented in Figure 7.

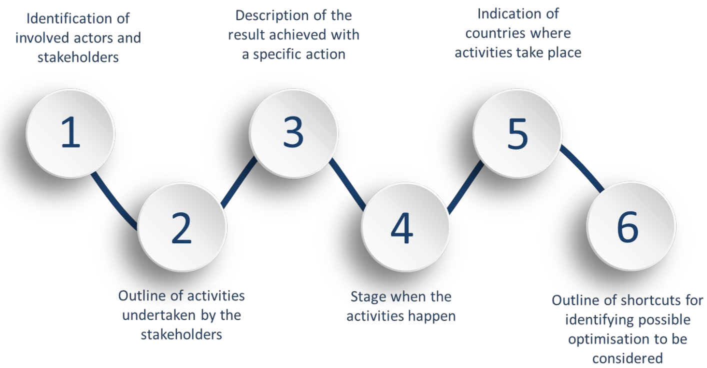

Figure 7: Main steps for analysing a renovation process (extracted from BIM4EEB D2.1)

3.1.2 Practical workability of ontologies

"***Operational means that the ontology specifications are available in a single, current ontology formalism for which scalable repositories, reasoning support, APIs, and tools are available***"

For the successful usability and operation of ontologies, BIM4EEB followed the OWL-2 DL profile in ontology development. The list of ontologies developed is shown in Table 6. Similarly, Table 5 clarifies the all set of commonly adopted practices for the development of BIM4EEB ontology suite. These practices are addressed with DiCon ontology suite in the section 4.1.2.

An online GitHub repository has been established for the maintenance and publication of the DiCon ontologies. The construction and workflow of this repository are illustrated through **Figure 8** below. These developed ontologies further uploaded to BIMMS environment to store the related knowledge bases and transfer to the BIM4EEB tools.

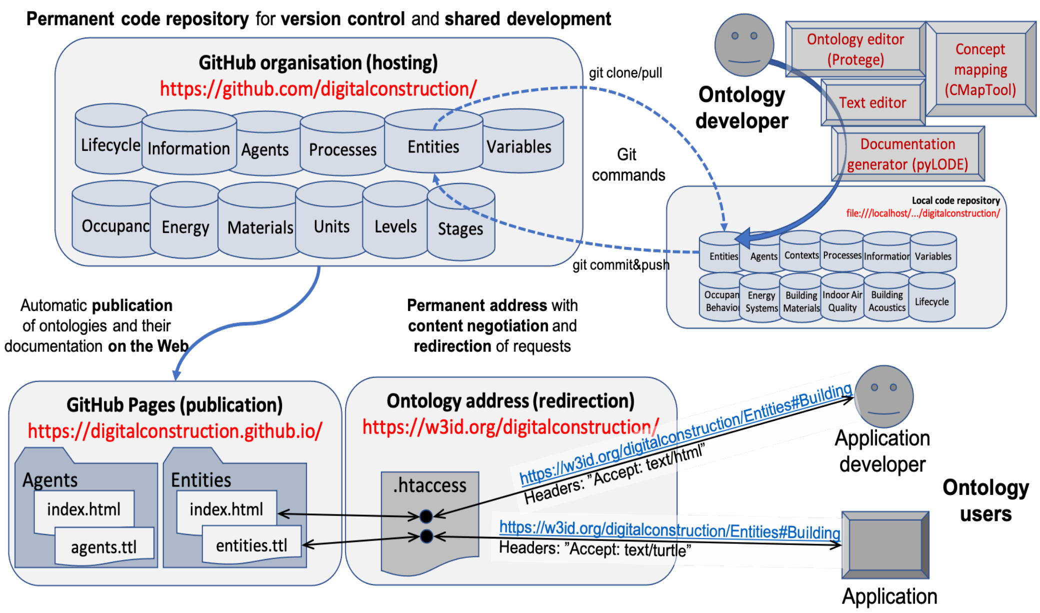

Figure 8: BIM4EEB Digital Construction Ontologies repository (BIM4EEB D3.1)

An important connection of the operational network ontologies is to the renovastion tools through their interfaces with the Linked data repositories. As illustrated in BIM4EEB framework in Figure 6, the toolset is a component highly dependent on BIMMS environment to receive and store data. Some of the tools have enough capabilities to store ontology related data and some fully dependent on interfaces like

SPARQL endpoint and REST API. This tool interface is represented in Figure 6 with information flow indications. Furthermore, the clear interaction between the BIM4EEB tools and the BIMMS are elaborated in Chapter 7 and also in Figure 24, Figure 25, Figure 26, and Figure 27.

3.1.3 Ontology consistency

"***Consistent means that the ontologies are based on compatible paradigms, have a compatible degree of detail, and include at least partial sets of alignment relations***."

With regarding the ***compatible paradigm***, BIM4EEB ontologies developed based on the Linked Data principles extensively defined by Tim Berners Lee (2006). An evaluation criterion is performed to BIM4EEB ontologies with the (open) Linked data principles and the results are clearly recorded in the following sections. The results suggest that the developed ontologies are highly compatible with Linked data principles.

Similarly, the ***compatible degree of ontology detail*** is evaluated through the concept of competency questions which are defined in BIM4EEB WP3 deliverables. These competency questions and the respective SPARQL catalogue is represented in chapter 6.

Finally, a set of ***alignments*** were developed as a part of ontology modularization in the chapter 5 and listed in Annex I. The alignment and interrelation of BIM4EEB ontologies with External ontologies is illustrated in section 5.2.

4 BIM4EEB ontologies, vocabularies, and external ontologies

The ontologies developed in the BIM4EEB project named Digital Construction Ontologies (DICon[2]). These ontologies will act as an enabler of semantic interoperability between the systems in the construction and renovation domain. Major areas taken to develop the ontologies are entities, information, static and dynamic properties, Level of Development (LoD), roles, renovation activities, energy efficiency, occupant comfort.

According to Deliverable D3.1 ontologies resemble scientific theories in the sense that they are never really finished. Instead, they can be refined, extended, and reformulated over time based on practical experiences. It means that an ontology should have an active community that learns, discusses, and applies it, and fixes bugs, identifies potential extensions, and revises the definitions.

For this reason, it was considered important to come up with an arrangement with BIM4EEB ontologies that would

(1) guarantee their continuing availability even after the time frame of the project;

(2) enable the continuing development of the ontologies by new parties and new projects.

The mechanisms used to achieve these objectives are the following:

(1) A permanent publication platform of github.com is used for the ontology source files and github.io for published files. A neutral identifier https://w3id.org/digitalconstruction has been allocated from W3C Perma-id (maintained by the W3C Permanent identifiers community group) to provide a continuing address for the ontologies. While nothing in this imperfect world is permanent, this arrangement can be regarded as the best effort to ensure the continuity of access to the ontologies.

(2) The ontologies have been licensed with Creative Commons Attribution license (CC BY 4.0) that makes the development by other groups easy. The neutral character of the ontologies and their naming facilitates the contributions from parties external to BIM4EEB.

(3) To ensure a broader community, the ontology definition effort has been initially combined with another research project, Diction (Business Finland, 2018-2020) focused on situational awareness in construction projects. There are active plans to utilize the ontologies also in new projects that are still at the proposal stage.

In the longer run, it is, of course, necessary to establish a body to coordinate the continuing development of the ontologies. At the minimum, there should be someone to review and accept pull requests to the ontology repositories in GitHub.

It needs to be determined whether some of the participants of BIM4EEB would assume this responsibility or if there is a continuation project that is willing to take that role.

[2] https://digitalconstruction.github.io/v/0.5/index.html

4.1 BIM4EEB Ontologies

4.1.1 Overview

The ontologies are published as an ontology suite. That is, there are several interrelated ontology modules, each focused on a particular theme. The ontologies developed in BIM4EEB are published on the site:

- **Development site:** https://github.com/digitalconstruction
- **Publication site:** https://digitalconstruction.github.io/
- **Permanent id:** https://w3id.org/digitalconstruction/

There are, as of now, ten different ontology modules. The Digital Construction Ontology Suite is shown in Table 6. Short descriptions of all ontologies in the ontology suite are provided. Only the general characteristics and the addresses of the ontologies are provided in the text.

The developed ontology suite is modularized. Some of the main classes considered in the BIM4EEB ontologies are illustrated in Figure 9 (overleaf).

Table 6: Digital Construction Ontology Suite (DICon)

Ontology Name	NameSpace	Description
Contexts	https://w3id.org/digitalconstruction/0.5/Contexts#	Multi-contexts data: planned/actual, as-designed/as-built, levels of detail
Variables	https://w3id.org/digitalconstruction/0.5/Variables#	Objectified properties for time-varying values, constraints, and value metadata
Entities	https://w3id.org/digitalconstruction/0.5/Entities#	Basic categories with identifiers, classifications, breakdowns, and groupings
Processes	https://w3id.org/digitalconstruction/0.5/Processes#	Activities and resources, resource assignments, and objects of activities
Agents	https://w3id.org/digitalconstruction/0.5/Agents#	Actors, stakeholders, roles, legal persons, capabilities, capacities
Information	https://w3id.org/digitalconstruction/0.5/Information#	Information content entities, information containers, designs, plans, events, and issues
Materials	https://w3id.org/digitalconstruction/0.5/Materials#	Building materials, material object structures, material properties, and material batches
Occupancy	https://w3id.org/digitalconstruction/0.5/Occupancy#	Occupant behavior, comfort, safety, and health; indoor air quality and building acoustics
Lifecycle	https://w3id.org/digitalconstruction/0.5/Lifecycle#	Evolution of information through LOD levels and over the construction lifecycle
Energy	https://w3id.org/digitalconstruction/0.5/Energy#	Energy efficiency including energy system

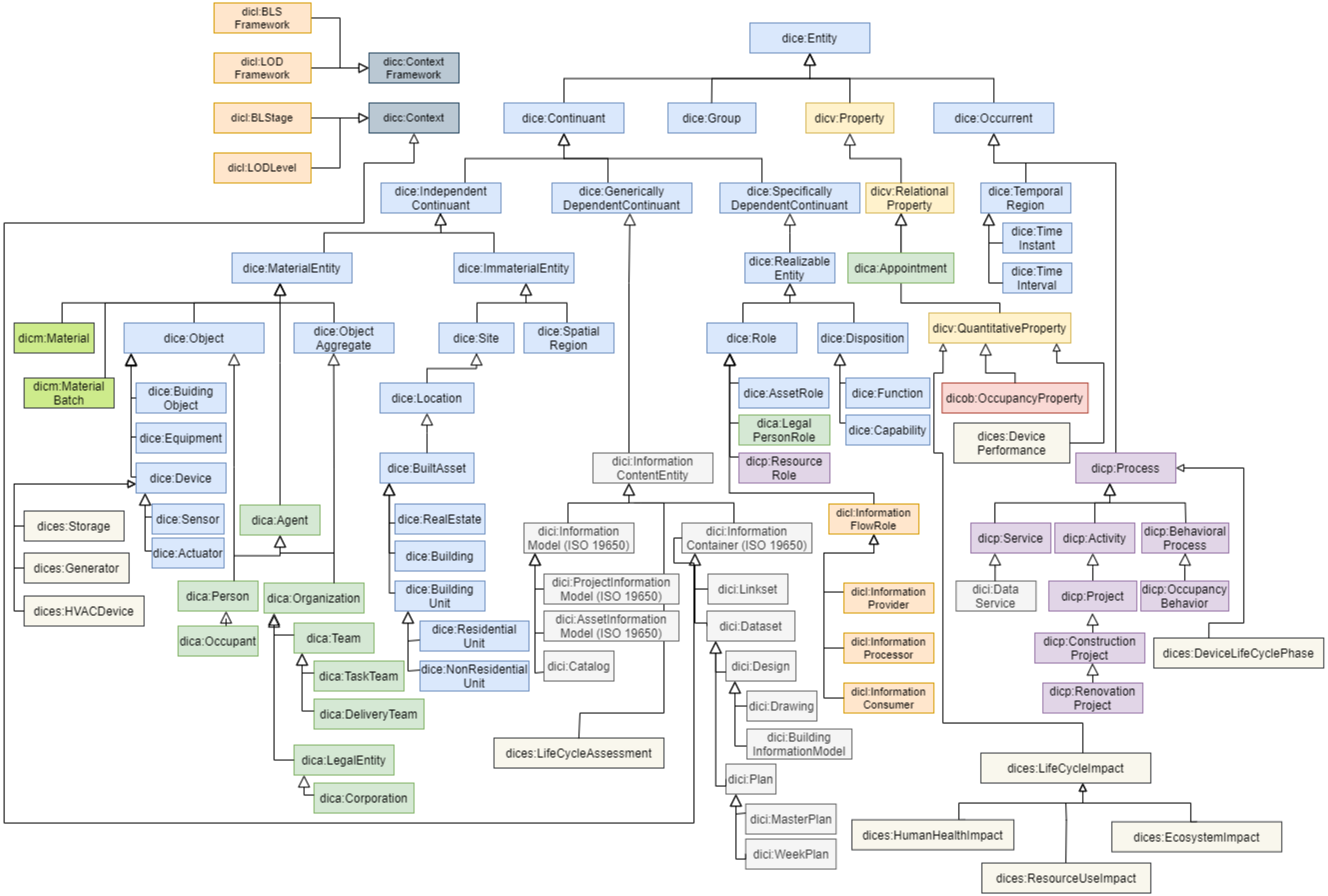

Figure 9: Overview of the main classes in the BIM4EEB ontologies

4.1.2 BIM-Ontologies: Specifications

4.1.2.1 Digital construction – Contexts

Defines the representation for multi-context data. How to represent both planned and actual values of same schedules (to be able to compute the deviations, for example), the designs in different stages of refinement such as as-designed/as-built designs or levels of detail/development (LOD), and the evolving positions of same objects at different time intervals. This model adopted in the Contexts ontology is based on the use of the graph-concept of RDF.

An RDF dataset is composed of a set of graphs, one of which is called the default graph and the other as named graphs. In the simplest case, all RDF content is stored in the default graph. However, it is possible to store it in any of the named graphs as well. For this reason, each RDF statement or a triple <subject, predicate, object> is internally represented as a quad <graph, subject, predicate, object>. Context ontology utilizes this built-in capability of RDF to record the context information of each triple. For instance, the actual and planned end time of an interval1 can be (in a slightly simplified manner) be recorded as:

- <plannedGraph, iv1, hasEnd, “2020-01-10”>
- <actualGraph, iv1, hasEnd, “2020-01-12”>

The advantages of this representation are

- the economy of metadata storage, since it uses a built-in capability of RDF,
- suitability for recording metadata of engineering-oriented data that is usually produced in large datasets – designs, plans, contracts, and so on – whose metadata is the same, and
- the orthogonality of context-dependent data, meaning that any data – for instance, breakdown structures or layout plans – can be context-dependent, and their evolution over time can therefore be recorded.

The practices and conventions considered in the framework are listed in Table 7.

Table 7: The information about Contexts ontology

Criteria	Implementation Data
Ontology Name	Digital Construction Contexts
Prefix	Dicc
NameSpace	https://w3id.org/digitalconstruction/0.5/Contexts#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Contexts#Context)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Contexts/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed	G4

4.1.2.2 *Digital Construction – Variables*

The values of most properties are subjected to change over the time of the renovation project. To capture the possibility of changes, all properties can be objectified and associated with different values at different times. Any object can be associated with objectified properties and the property state with the object property. The use of property states makes it possible to associate properties with different values at different times and coming from different origins. The classes are defined in the variables ontology for the objectification of quantitative properties. They allow the connection of a quantity kind to a property as well as the unit of measurement. Table 8 lists the conventions and practices considered for the Variables ontology.

Table 8: The information about Variables ontology

Criteria	**Implementation Data**
Ontology Name	Digital Construction Variables
Prefix	Dicv
NameSpace	https://w3id.org/digitalconstruction/0.5/Variables#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Variables#CompositeConstraint)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Variables/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed	G5

4.1.2.3 *Digital Construction – Entities*

The upper structure of the ontology is defined in the Entities; it is based on BFO and aligned with it. The entities module also extends BFO with concepts that are specific to the construction and renovation domain. Defines the main entity classes in the building domain. For instance, it defines an asset role and using it the concept of a built asset to comply with the parlance of ISO 19650. The subclasses of a built asset are real estate, building, and building units.

The central concept is an Identifiable Entity that can have:

- identifiers: each entity can have multiple different identifiers that can be either globally unique or unique in some local context, such as room numbers of a building;
- classifications: each entity can be associated with classification codes based on different classification systems, such as Omniclass, ETIM, CoClass, Talo2000, and so on;
- breakdowns: each entity can be decomposed into smaller parts.

The practices and conventions used in the Entities ontology are listed in Table 9.

Table 9: The information about Entities ontology

Criteria	Implementation Data
Ontology Name	Digital Construction Entities
Prefix	Dice
NameSpace	https://w3id.org/digitalconstruction/0.5/Entities#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Entities#Entity)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Entities/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed in the ontology	G1

4.1.2.4 Digital construction – Processes

The process ontology defines the activities, resources, and resource role. The concept of Resource Role determines what is a resource. To be a resource is not the inherent characteristic of an entity but a role that different entities can play at different times concerning activities. Any entity that has a resource role can be classified as a resource. Resource role is associated with activity through the relation. The resource role is thus concerning an activity. The resource role can further be specified with the datatype property indicating the amount the activity uses the resource. The resource has a unit cost, which enables the computation of the cost of the resource use in the activity.

The practices and conventions used in the Entities ontology are listed in Table 10.

Table 10: The information about Processes ontology

Criteria	Implementation Data
Ontology Name	Digital Construction Processes
Prefix	dicp
NameSpace	https://w3id.org/digitalconstruction/0.5/Processes#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Processes#Activity)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Processes/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed	G11

4.1.2.5 Digital construction – Agents

Defines the classes for agents, such as persons and organizations. The purpose is to represent the actors and stakeholders over the construction lifecycle. Actors are agents that do activities, and stakeholders are agents that hold interests in the project or real estate. Practices and conventions are listed in Table 11.

Table 11: The information about Agents ontology

Criteria	**Implementation Data**
Ontology Name	Digital Construction Agents
Prefix	dica
NameSpace	https://w3id.org/digitalconstruction/0.5/Agents#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Agents/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed	G2

4.1.2.6 Digital construction - Information

Defines core information objects such as design, plan, contract, issue, and event. The ontology focuses on the information content, not physical information carriers (e.g. paper or data storage). ISO 19650 defines the practices for “information management using building information modeling”. The main concepts are the Information Model and Information Container. Information Container corresponds roughly to a file containing a dataset of some kind. The Information Model is a set of Information Containers organized hierarchically to a so-called federation or information container breakdown structure. These concepts are considered in the Information ontology. Practices and conventions are listed in Table 12.

Table 12: The information about Information ontology

Criteria	**Implementation Data**
Ontology Name	Digital Construction Information
Prefix	dici
NameSpace	https://w3id.org/digitalconstruction/0.5/Information#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Information/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed	G3

4.1.2.7 Digital construction –Materials

The Materials ontology defines the main concepts of building materials, types, Material object structure, and properties. The order and position of layers in building elements concerning other elements are efficiently defined. The practices and conventions used in the Entities ontology are listed in Table 13.

Table 13: The information about Materials ontology

Criteria	**Implementation Data**
Ontology Name	Digital Construction Materials
Prefix	dicm
NameSpace	https://w3id.org/digitalconstruction/0.5/Materials#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Materials#Material)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Materials/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed in the ontology	G9

4.1.2.8 Digital construction – Occupancy

Defines the representation for describing the occupants' behavior and comfort inside a building. For that, the ontology models the environment and users' interaction to set the parameters for comfort behavioral profile. The objective is to capture the dynamic occupancy and visual/thermal comfort models to facilitate the definition of accurate comfort profiles for occupants in buildings.

Defines an operational framework for the orchestration of heterogeneous data related to the environment, health, and user profiles, to support user-oriented behavioral profiles. Also, Defines a representation for building acoustics covering concepts such as Airborne Sound Improvement, Airborne Sound Insulation, Flanking Sound Transmission, Impact Noise, Joint Sound Insulation, Sound Absorption, and Vibration Reduction Index. The practices and conventions used in the Entities ontology are listed in Table 14.

Table 14: The information about Occupancy ontology

Criteria	Implementation Data
Ontology Name	Digital Construction Occupancy
Prefix	dicob
NameSpace	https://w3id.org/digitalconstruction/0.5/Occupancy#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Occupancy#OccupantBehavior)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Occupancy/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed in the ontology	G6, G7

4.1.2.9 Digital construction – Lifecycle

The lifecycle ontology represents the enhancement of building data throughout the construction lifecycle stages. The practices and conventions used in the Entities ontology are listed in Table 15.

Table 15: The information about Lifecycle ontology

Criteria	Implementation Data
Ontology Name	Digital Construction Lifecycle
Prefix	dicl
NameSpace	https://w3id.org/digitalconstruction/0.5/Lifecycle#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Lifecycle#BuildingLifecycleStage)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Lifecycle/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed in the ontology	G10

4.1.2.10 Digital construction – Energy

An extension of SAREF to address lifecycle assessment (LCA) parameters for the energy systems to be considered in a renovation project. The practices and conventions used in the Entities ontology are listed in Table 16.

Table 16: The information about Energy ontology

Criteria	**Implementation Data**
Ontology Name	Digital Construction Energy
Prefix	dices
NameSpace	https://w3id.org/digitalconstruction/0.5/Energy#
Profile	OWL2 DL
Identifiers	HTTP URIs, using the HTTPS protocol (example: https://w3id.org/digitalconstruction/0.5/Energy#EnergyConsumptionQuality)
Data format	Turtle
License	https://creativecommons.org/licenses/by/4.0/
Hosting	https://w3id.org/digitalconstruction
Documentation	Using pyLODE
Publication	https://digitalconstruction.github.io/Energy/v/0.5/
Versioning	https://digitalconstruction.github.io/v/0.5/
Conceptual gaps addressed in the ontology	G8

4.2 BIM4EEB Vocabularies

The vocabularies are developed for the units, information levels, building lifecycle stages. The developed vocabularies are published in the GitHub webpage “https://digitalconstruction.github.io/v/0.5/”.

Table 17: Digital Construction Vocabulary

Vocabulary Name	**Prefix**	**NameSpace**	**Description**
Units	dicu:	https://w3id.org/digitalconstruction/0.5/Units#	The vocabulary of units and quantity kinds specific to construction and renovation
Levels	diclvl:	https://w3id.org/digitalconstruction/0.5/Levels#	The vocabulary of LOD levels based on established frameworks in EU, IT, UK, and USA
Stages	dicstg:	https://w3id.org/digitalconstruction/0.5/Stages#	The vocabulary of lifecycle stages based on the frameworks of ISO 22263, RIBA, and HOAI

4.3 External ontologies used in the BIM4EEB project

As already described in Deliverable D3.1 numerous ontologies were analyzed. Compared to ifcOWL the expressivity and complexity of these ontologies are much lower. Out of the total list of analyzed ontologies, we selected the 16 ontologies to progress with. These are listed in Table 18. A short textual specification follows on the next page.

Table 18: Ontologies re-used in BIM4EEB as a whole or parts

No.	Akronym	Full Name	URL	Prefix	NameSpace
1	BFO	Basic Formal Ontology	https://basic-formal-ontology.org/	bfo	http://purl.obolibrary.org/obo/bfo.owl#
2	FOAF	Friend of a Friend	http://xmlns.com/foaf/spec/	foaf	http://xmlns.com/foaf/0.1/
3	QUDT	Quantity, Units, Dimen- sions & Types Schema	http://www.qudt.org/2.1/catalog/qudt-catalog.html	qudt	http://qudt.org/2.1/schema/qudt
4	OWL-TIME	Time ontology in OWL	https://www.w3.org/TR/owl-time/	time	http://www.w3.org/2006/time
5	PROV-O	Provenance Ontology	https://www.w3.org/TR/prov-o/	prov	http://www.w3.org/ns/prov#
6	SSN	Semantic Sensor Networks	https://www.w3.org/TR/vocab-ssn/	ssn	http://www.w3.org/ns/ssn/
7	SOSA	Sensor, Observation, Sample and Actuator	https://www.w3.org/TR/vocab-ssn/	sosa	http://www.w3.org/ns/sosa/
8	SAREF	Smart Appliances REFerence	https://saref.etsi.org/core/v3.1.1/	saref	https://w3id.org/saref#
9	SAREF 4BLDG	SAREF ontology for Buildings	https://saref.etsi.org/saref4bldg/v1.1.2/	s4bldg	https://w3id.org/def/saref4bldg#
10	BOT	Building Topology Ontology	https://w3c-lbd-cg.github.io/bot/	bot	https://w3id.org/bot#
11	ifcOWL	IFC ontology	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL/ontology.ttl	ifc	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#
12	OPM	Ontology for property management	https://w3c-lbd-cg.github.io/opm/	opm	http://www.w3id.org/opm#
13	RealEstate Core	Real Estate Core ontology	https://www.realestatecore.io/download	core	https://w3id.org/rec/core/
14	Org	Organization Ontology	https://www.w3.org/TR/vocab-org/	org	http://www.w3.org/ns/org#
15	DCAT	Data Catalog Vocabulary	https://www.w3.org/TR/vocab-dcat-2/	dcat	http://www.w3.org/ns/dcat#
16	VoID	Vocabulary of Interlinked Datasets	http://vocab.deri.ie/void	void	http://vocab.deri.ie/void
17	CT	Container Ontology	https://standards.iso.org/iso/21597/-1/ed-1/en/Container.rdf	ct	https://standards.iso.org/iso/21597/-1/ed-1/en/Container#
18	Wgs84_pos	WGS84 Geo Positioning: an RDF vocabulary	https://www.w3.org/2003/01/geo/	geo	http://www.w3.org/2003/01/geo/wgs84_pos#

The Basic Formal Ontology (bfo[3]) is an upper-level ontology used in supporting information retrieval, analysis, and integration in scientific domains. The Friend of a Friend (foaf[4]) ontology defines the terms related to the people. The Quantities, Units, Dimensions, and Types (qudt[5]) ontology defines the terms related to the quantities, quantity kinds, units, dimensions, and data types. OWL-Time (time[6]) is an ontology of temporal concepts, for describing the temporal properties of resources.

The PROV Ontology (prov[7]) provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. The Organization Ontology (org[8]) ontology for organizational structures, aimed at supporting linked data publishing of organizational information across several domains.

The Semantic Sensor Network Ontology (ssn9) ontology describes sensors, the accuracy and capabilities of sensors, observations, and methods used for sensing. Concepts for operating and survival ranges are included, along with its performance within those ranges [Compton,2012]. The Sensor, Observation, Sample, and Actuator (sosa[9]) ontology describe elementary classes and properties of these items.

The Smart Applications REFerence Ontology (saref[10]) ontology is intended to enable interoperability between solutions from different providers and among various activity sectors in the Internet of Things (IoT). The SAREF extension for building (s4bldg[11]) ontology extends the SAREF ontology for the building domain by defining building devices and how they are located in a building.

Ifcowl (ifc[12]) is the ontology developed based on the IFC schema. It mainly defines the terms related to the building models, construction resources, project contexts, and property sets information.

The Building Topology (bot[13]) ontology explicitly defines the terms and relationships between the subcomponent of the building [Rasmussen,2019].

The Ontology for Property Management (opm[14]) is an ontology for describing temporal properties that are subject to changes as the building design evolves. The RealEstateCore (core[15]) is an ontology to make buildings interact with the smart city. Data Catalog Vocabulary (dcat[16]) is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. The Vocabulary of Interlinked Datasets (void[17]) is an RDF Schema vocabulary for expressing metadata about RDF datasets. The Container (ct[18]) ontology is an ontology providing the object classes and properties that shall be used to specify the contents of a container [ISO 21597-1]. The WGS84 (geo[19]) Geo Positioning is an RDF vocabulary representing latitude, longitude, and altitude information in the WGS84 geodetic reference datum.

---

[3] https://basic-formal-ontology.org/
[4] http://xmlns.com/foaf/spec/
[5] http://www.qudt.org/pages/QUDToverviewPage.html
[6] https://www.w3.org/TR/owl-time/
[7] https://www.w3.org/TR/prov-o/
[8] https://www.w3.org/TR/vocab-org/
[9] https://www.w3.org/TR/vocab-ssn/
[10] https://saref.etsi.org/core/v3.1.1/
[11] https://saref.etsi.org/saref4bldg/v1.1.2/
[12] https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL/ontology.ttl
[13] https://w3c-lbd-cg.github.io/bot/
[14] https://w3c-lbd-cg.github.io/opm/
[15] https://www.realestatecore.io/download
[16] https://www.w3.org/TR/vocab-dcat-2/
[17] http://vocab.deri.ie/void
[18] https://standards.iso.org/iso/21597/-1/ed-1/en/Container.rdf
[19] https://www.w3.org/2003/01/geo/

5 Ontologies modularization

Modularization of ontologies makes it easier for users to understand, extend, reuse, maintain, and reason ontologies [Bernardo,2007] [Mathieu,2009]. However, the concept of modularization is not well defined in the context of ontologies compared to software engineering. A single approach for modularization does not match every situation since people tend to have various ideas in the development of ontologies.

Several approaches appeared in the field of ontology modularization. These approaches are mainly categorized into “ontology separation” and “ontologies composition” and are shown in Figure 10. These two main approaches are sub-categorized into ontology partition, ontology module extraction, ontologies integration, and ontologies mapping respectively [Sarra,2012].

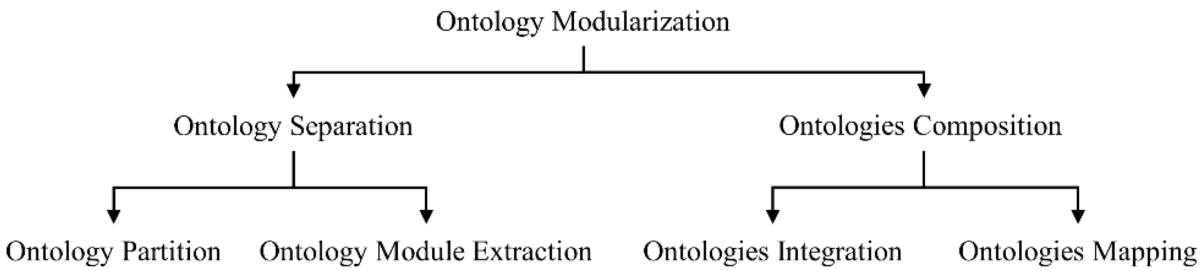

Figure 10: Ontology modularization approaches

In the BIM4EEB project, Digital Construction Ontologies (DICon[2]) are developed to achieve semantic interoperability and enhance the information sharing and representation of renovation data in the building renovation life cycle process. To expand the scope of ontologies usage, relations are established between the DICon and external ontologies by using ontology modularization.

5.1 Principles for Ontology Integration

Ontology integration is the process of forming a new ontology by using one or more ontologies without changing their original concepts, if possible they are extended [Marc, 2007].

Integrate (O1, O2, A) = O1, where O1 is the target ontology into which the source ontology O2 will be integrated and A is the alignment expressed in the same logical language as ontologies O1 and O2 [Inès, 2021]. Ontology alignment may be seen as a pre-step for detecting where the involved ontologies overlap and can be connected. This approach is especially interesting if given ontologies differ in their domain. Through integration, the new ontology can cover a bigger domain in the end. Based on the approach used by the Semantic Sensor Network Ontology[9], the ontology integration process is categorized into vertical and horizontal segmentation.

For the vertical and horizontal segmentation, the principles described in the deliverable “D3.1 A BIM-based framework for building renovation using the linked data approach and ontologies–State-of-the-art, use cases, and high-level architectural specifications”. This segmentation process elaborately described in the following sections.

5.1.1 Horizontal Segmentation:

The ontologies can be segmented horizontally as explained in the D3.1. The horizontal segmentation process is explained more clearly in Figure 11.

Horizontally layered modules depend on the directional import of another module. The ontologies O1 and O2 conceptually represent BIM4EEB ontologies. To establish a connection between the ontologies, as an initial step the overlapping concepts between ontologies are removed. For example, O1:C1 is a class in O1 and the same class is defined in the ontology O2 with the URI of O1. In the second step, ontology O1 is imported to ontology O2.

This approach helps to avoid redundancy and the ontology merging process will be easier.

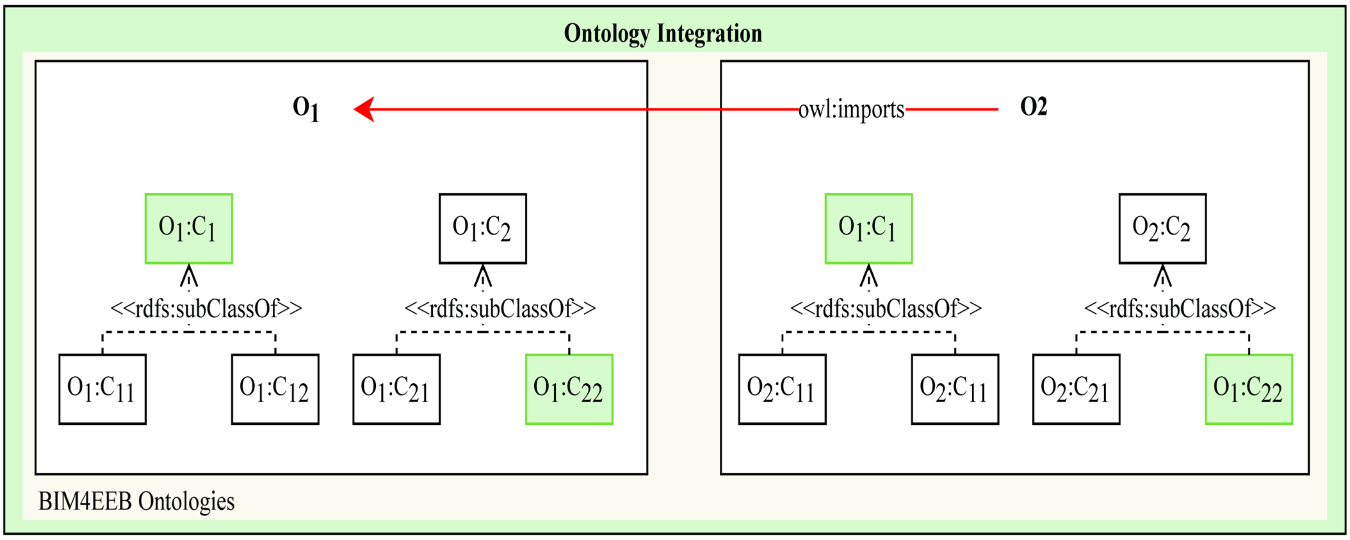

Figure 11: Horizontal segmentation for ontologies modularization

5.1.2 Vertical Segmentation:

The ontologies can be segmented vertically as explained in the D3.1. The vertical segmentation process is explained in Figure 12. The O_1 and O_2 conceptually represent BIM4EEB ontologies and O_3 represents an external ontology. O_1-O_3 and O_2-O_3 are the alignment modules. To establish a connection between the ontologies a two-step approach is deployed. In the first step, alignment modules are developed between the BIM4EEB ontologies and external ontologies and kept as a separate file. Ontology alignment is always based on vertical segmentation. In the second step, All external ontology references are made explicit through separate alignment modules.
However, the above approach does not apply to references to external vocabularies (instances) that will be allowed in the ontologies themselves. In other terms, the alignment module will

(1) Import all the ontologies to be aligned;
(2) Define additional axioms that refer to the terms defined in the aligned ontologies.

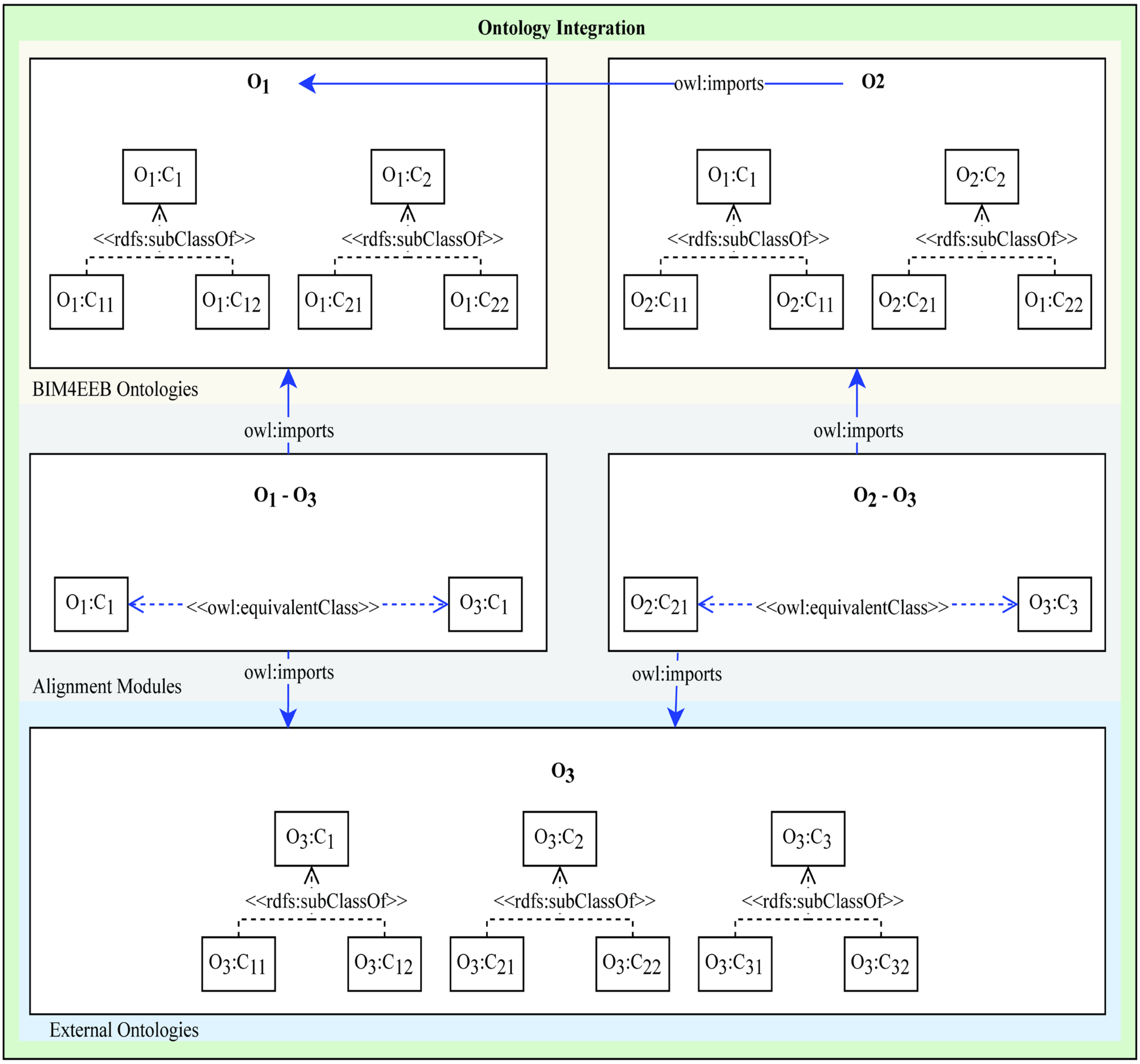

Figure 12: Vertical segmentation for ontology modularization

5.2 Ontology Integration in BIM4EEB

5.2.1 Alignment between the context and external ontologies

The concept dicc:Context is an identified realm of data, representing the circumstances in which the data can be considered true. On the other hand ifc:IfcContext is the generalization of a project context in which objects, type objects, property sets, and properties are defined [ISO16739,2013]. These two concepts are terminologically related to each other. The concept dici:Context represents the context information more general than the ifc:IfcContext. So, ifc:IfcContext is a subClassOf dicc:Context. The alignment is shown in Table 93.

Figure 13 illustrates the integration of contexts ontology (dicc) with external ontologies by using the alignment modules. The alignments between the ifc and contexts ontology developed in a separate file and both contexts and ifc ontologies are imported into it by using vertical segmentation.

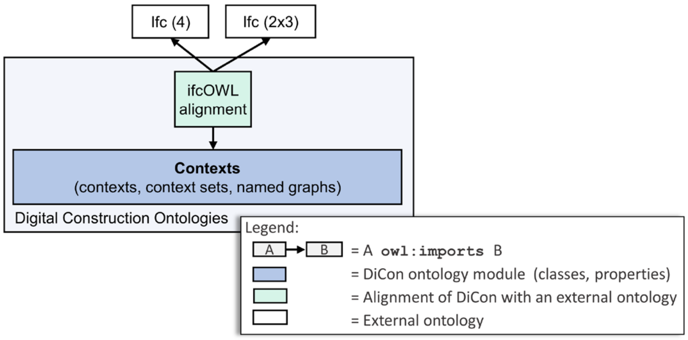

Figure 13: Integration of other ontologies into DICC ontology

5.2.2 Alignment between the variables ontology with external ontology

The variables ontology dicv is aligned with qudt, opm, ssn, saref, sosa,ifc ontologies. The concepts dicv:QuantityKind, dicv:Unit are the same concepts as defined in the qudt ontology. The class dicv:QuantityKind is any observable property that can be measured and quantified numerically. Familiar examples include physical properties such as length, mass, time, force, energy, power, electric charge, etc. Less familiar examples include currency, interest rate, price to earnings ratio, and information capacity.

The qudt:Unit is a unit of measure. The classes dicv:PropertyState, dicv:Property, dicv:QuantitativeProperty are aligned with the classes opm:PropertyState, opm:Property, ssn:Property and saref:Property. The concept dicv:Property is an objectified property and dicv:PropertyState defines or constraints the value of a property. The class The class dicv:QuantitativeProperty is a property that can assume quantitative values. The sosa:ObservableProperty is an observable quality (property, characteristic) of a FeatureOfInterest and it is aligned with dicv:Property. Saref:UnitOfMeasure is a standard for measurement of a quantity and aligned with dicv:Unit. The object properties in the variables ontology are aligned with qudt, opm, saref ontologies as shown in Table 94.

The integration of variables ontology with external ontology is shown in Figure 14. The alignment modules developed between the Variables ontology and external ontologies saref, prov, ssn/sosa, opm, qudt, ifc. The aligned modules import the ontologies using the vertical segmentation approach.

In Figure 14 “A” represents the alignment file and “B” represents variables and external ontologies.

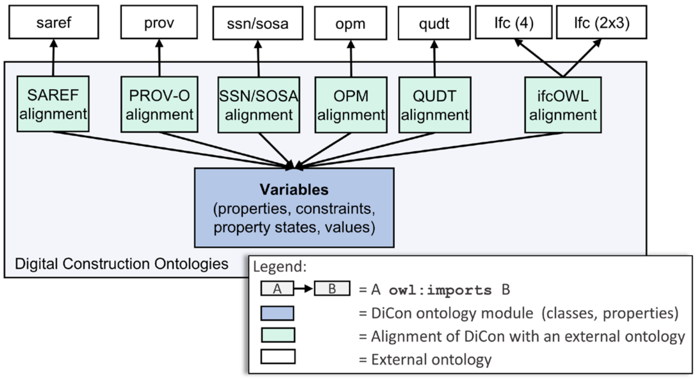

Figure 14: Integration of other ontologies into DICV ontology

5.2.3 Alignment between entities ontology with other ontologies

The dice ontology is aligned with the ifc, saref, bot, s4bldg, prov, foaf, org, bfo, time, ct, ssn/sosa, core, and geo ontologies. The Ifc:IfcElement represents some concepts of dice:Object. The ifc:IfcSpatialElement, bot:Zone, s4bldg:BuildingSpace are terminologically the same as dice:Location. The dice:Location is a place where material entities can be located or activities can occur.

The dice:Sensor is a device that is capable of producing information about its environment. The ifc:IfcSensor, saref:Sensor are terminologically, semantically the same as dice:Sensor. The dice:Group represents a set of entities and it is equivalent to the concept ifc:IfcGroup. The dice:Building is the same as bot:Building in terms of terminology. The dice:Role represents a person's role in the construction project and it is aligned with org:Role. The entities ontology aligned with entities ontology is listed in Table 95 and Table 96.

Table 96 The entities ontology integrates with BIM4EEB and external ontologies as shown in Figure 15. Vertical segmentation is used for the integration process.

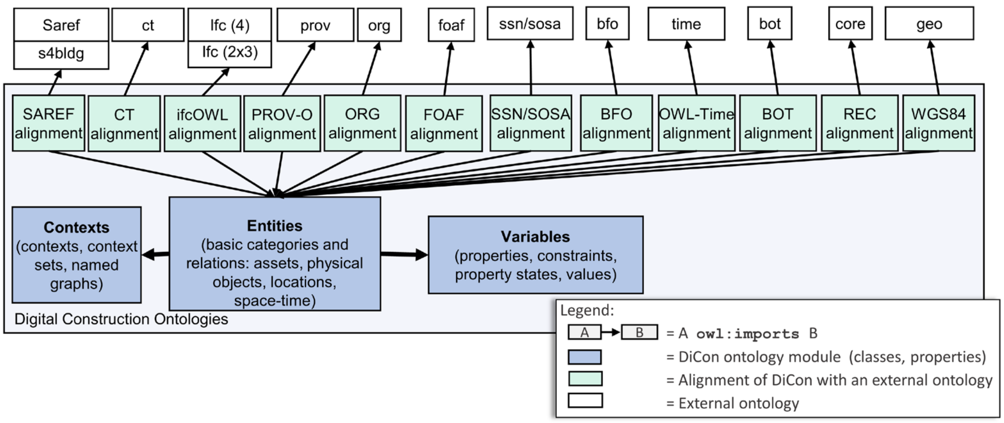

Figure 15: Integration of other ontologies into DICE ontology

5.2.4 Alignment between processes ontology with other ontologies

The concept dicp:Resource is any entity that plays a ResourceRole. The ifc:IfcResource, ifc:IfcConstructionResource is aligned with dicp:Resource. The dicp:Project is a ObjectActivity with specific goals and aligned with ifc:IfcProject. The dicp:Observation is a process of capturing information about some property of a feature of interest. The sosa:Observation is aligned with dicp:Observation. The dicp:Actuation is a process of acting on some property of a feature of interest and this concept is aligned with sosa:Actuation. The dicp:Service is a process in which service providers perform functions in response to requests of service requestors. The saref:Service is same as dicp:Service. The dicp:Activity is a process that is intentionally performed by an agent and aligned with prov:Activity. The few dicp object properties are aligned with sosa object properties and are listed in Table 97.

The integration of Processes ontology is integrated with BIM4EEB and external ontologies. The Processes ontology imports entities ontology and also, contexts and variables ontology. The horizontal segmentation approach is used for this. By using the vertical segmentation approach, Processes ontology integrates with saref, ifc, prov, foaf, sosa, and core ontologies. Figure 16 represents the integration of Processes ontology with other ontologies

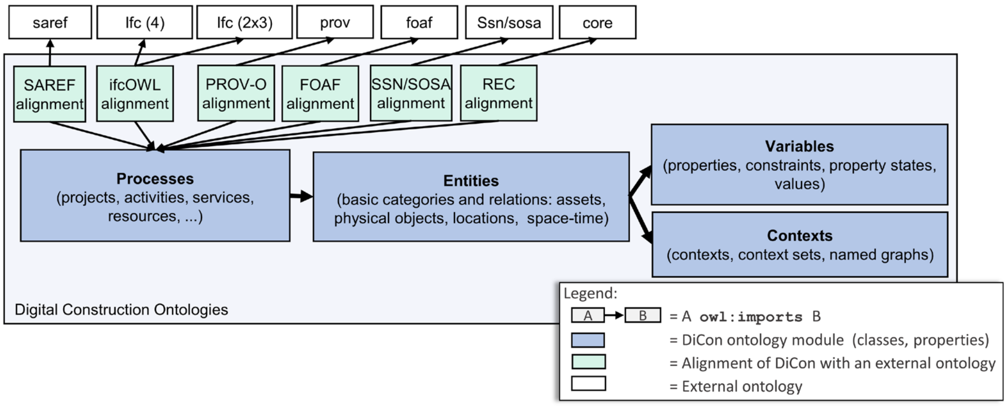

Figure 16: Integration of other ontologies into DICP ontology

5.2.5 Alignment between the agents and external ontologies

The dica:Actor is an agent that performs activities and is aligned with ifc:IfcActor. The dica:Person is a human being involved in the construction process. The ifc:IfcPerson, foaf:Person are aligned with the concept dica:Person. The dica:Organization is an organized group of people with a particular purpose involved in the process. The ifc:IfcOrganization, foaf:Organization, org:Organization is aligned with the dica:Organization. The dica:Occupant is a person that is occupant in some built asset. The ifc:IfcOccupant is equivalent to the dica:Occupant. The dica:Agent is an actor or stakeholder associated with construction lifecycles. The ct:Party, foaf:Agent is aligned with the dica:Agent. The dica:Team is a set of persons working together for some purpose and this concept is aligned with the foaf:Group. The dica:LegalEntity is an organization that has legal rights and responsibilities in the applicable jurisdiction. The concept org:FormalOrganization is a dica:LegalEntity. The dica concepts are aligned with the ontologies ifc, ct, foaf, org ontologies as shown in Table 98. The agent's ontology is integrated with entities, container, ifc, org, foaf, and core ontologies using horizontal and vertical segmentation approach. The process of integration is shown in Figure 17.

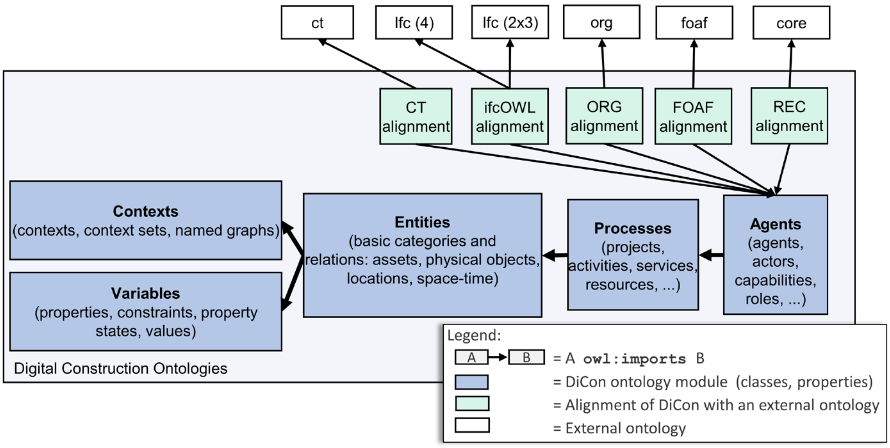

Figure 17: Integration of other ontologies into DICA ontology

5.2.6 Alignment between the information and external ontologies

The dici:InformationContentEntity is a identifiable information content. It is aligned with ifc:IfcApproval, ifc:IfcPermit, foaf:Document, and prov:Entity by taking its terminological descriptions and semantics. The dici:Plan represents set of activities with constraints associated with them. The concept dici:Plan is aligned with ifc:IfcWorkPlan, ifc:IfcWorkSchedule.The Dici:Event is a occurence that happens with a time instant for obtaining the information and it is similar to Ifc:IfcEvent. The dici:InformationModel is a set of structured and unstructured information containers (ISO 19650-1:2018). The possible alignment is with the concept ct:ContainerDescription. The dici:CrossFileLinkset is consisting of links between two or more datasets and is aligned with the concept ct:Linkset. The dici:Image is an information content entity containing visual data and it is the same as foaf:Image. All possible alignments with external ontologies are listed in Table 99.

The information ontology imports agents ontology by using the horizontal approach. Which eventually imports processes, entities, variables, and contexts ontologies indirectly. The container, ifc, prov, foaf, dcat, and core ontologies are integrated with information ontology using the vertical segmentation approach. The integration process is shown in Figure 18.

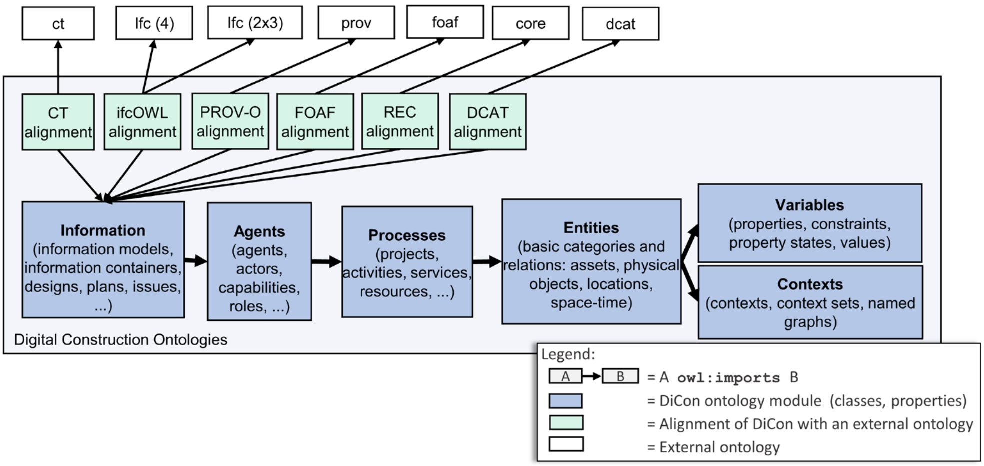

Figure 18: Integration of other ontologies into DICI ontology

5.2.7 Alignment between the material ontology and external ontologies

The referenced schema for the dicm ontology development is IFC. The major classes dicm:Material, dicm:MaterialObjectStructure, dicm:Layer, dicm:LayerSet is aligned with the concepts in the ifc ontology ifc:IfcMaterial, ifc:IfcMaterialDefiniton, ifc:IfcMaterialLayer, ifc:IfcMaterialLayerSet. The concept dicm:Material a homogeneous or inhomogeneous substance that can be used to form elements (physical products or their components). The concept dicm:MaterialObjectStructure is a concept to represent material-related information that has material-related properties. The class dicm:Layer is a single and identifiable part of an element that is constructed of several layers (one or more). The class dicm:LayerSet a designation by which materials of an element constructed of several material layers are known and through which the relative positioning of individual layers can be expressed. The object properties dicm:hasMaterial, dicm:adjacentElement are aligned with Ifcowl: material_IfcMaterialLayer, bot:adjacentElement. They terminologically represent the same concept.

The material ontology is integrated with the other BIM4EEB ontologies and external ontologies using horizontal and vertical segmentation approach and it is shown in Figure 19. Using the vertical segmentation materials ontology integrates with ifc and bot ontology. Using horizontal segmentation, materials ontology integrates with Entities and it also imports variables and contexts ontology.The dicm ontology aligned with the concepts and properties defined in the ifc ontology and bot as shown in Table 100.

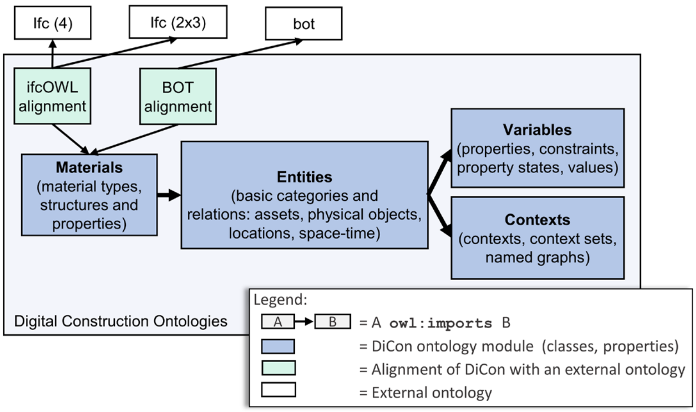

Figure 19: Integration of other ontologies into DICM ontology

5.2.8 Alignment between the occupancy ontology and external ontologies

The classes dicob:AirQualitySensor, dicob:HumiditySensor, dicob:LuminanceSensor, dicob:NoiseSensor, dicob:TemperatureSensor are subclass of Ifc:IfcSensor, sosa:Sensor, saref:Sensor, core:Sensor. A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument (IFC). AirQuality Sensor, Humidity Sensor, Luminance Sensor, Noise Sensor, and Temperature Sensor are used to measure the levels of air quality, humidity, luminance, noise, the temperature in a zone respectively. The dicob:Ceiling is a covering used to represent a ceiling and aligned with ifc:IfcCovering. The dicob:Door, dicob:Floor, dicob:FloorCovering, dicob:Covering is aligned with Ifc:IfcDoor, Ifc:IfcSlab, bot:Element, Ifc:IfcCovering, Ifc:IfcWall respectively and also aligned with bot:Element, core:BuildingComponent.

The Occupancy ontology (dicob) is integrated with Information, Agents, Processes, Entities, Contexts, and Variables ontologies using horizontal segmentation. By using vertical segmentation, dicob is integrated with ifc, saref, rec, sosa, and bot ontologies. The ontology integration process is shown in Figure 20. The Occupancy ontology (dicob) is aligned with ifc, sosa, saref, core, and bot ontologies and listed in Table 101.

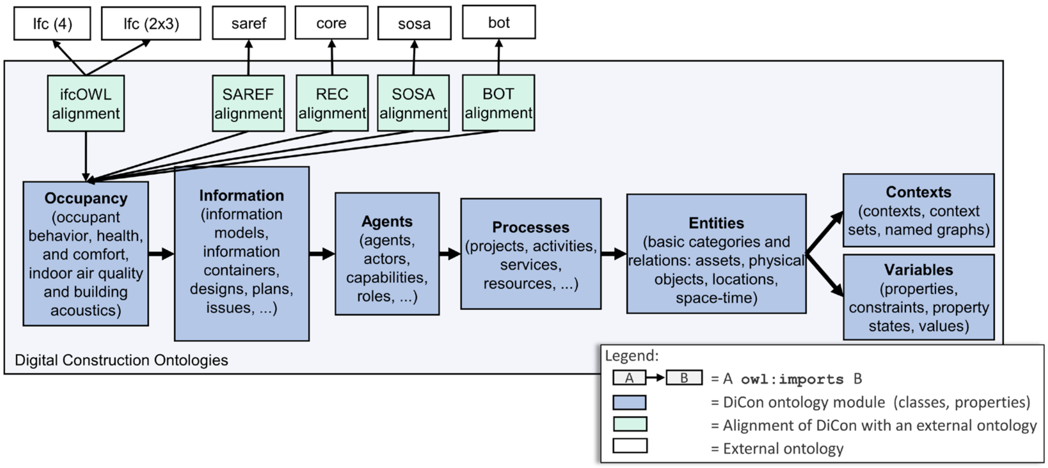

Figure 20: Integration of other ontologies into DICOB ontology

5.2.9 Alignment between the lifecycle ontology and external ontologies

The dicl:InformationConcumer is an agent who consumes and uses information (models, drawings, other datasets) from activity. The dicl:InformationProvider is an agent who processes, updates, and manages information (models, drawings, other datasets) to/for activity.The dicl:InformationProvider is an agent who provides information (models, drawings, other datasets) to/for activity. The concept dicl:InformationFlowRole is a role performed by an actor in the information exchange and processing. All these classes are aligned with the ifc, org ontology classes by taking the terminologies, semantics, and taxonomy into consideration. The dicl ontology aligned with the ifc, org ontology as shown in Table 102.

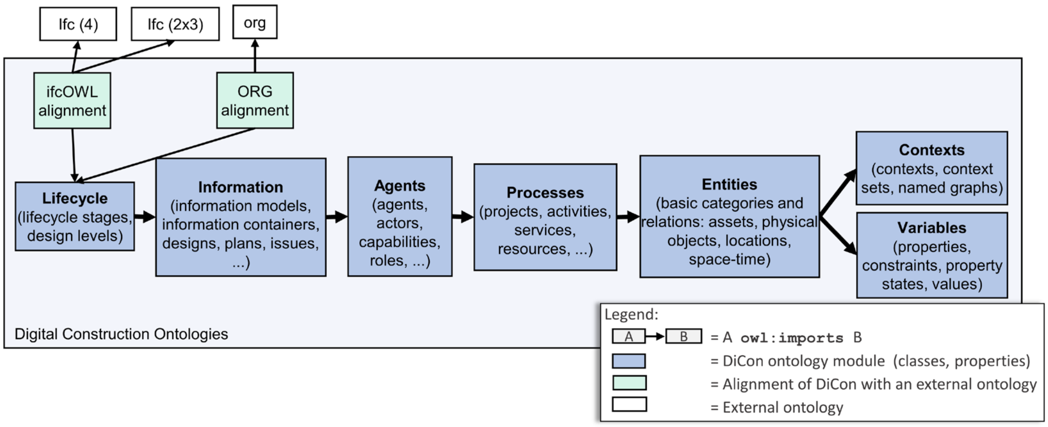

Figure 21: Integration of other ontologies into DICL ontology

The Lifecycle ontology (dicl) is integrated with the other ontologies using the horizontal and vertical segmentation approach as shown in Figure 21. Information ontology is imported into the Lifecycle ontology and it is a direct import and it is horizontal import. Since Information ontology imports other BIM4EEB ontologies Agents,Processes, Entities, Contexts, and Variables directly and indirectly. These all ontologies will be indirectly imported into the Life Cycle ontology. Also, Lifecycle ontology integrated with ifc and org ontologies using vertical segmentation approach.

5.2.10 Alignment between the energy ontology and external ontologies

The dices ontology aligned with the ifc and saref ontologies as shown in Table 103. The concepts dices:Actuator, Ifc:IfcActuator, saref:Actuator are terminologically correct and they represent a mechanical device that controls a mechanism. An actuator is a mechanical device for moving or controlling a mechanism or system [IFC]. Ifc:IfcElectricGenerator is a type of generator and its superclass will be dices:Generator.

The Energy ontology imports information ontology directly and agents, Processes, Entities, Contexts, and Variables ontologies indirectly by using vertical and horizontal segmentation. By using vertical segmentation Energy ontology integrates with ifc and saref ontologies. The integration process is shown in Figure 22.

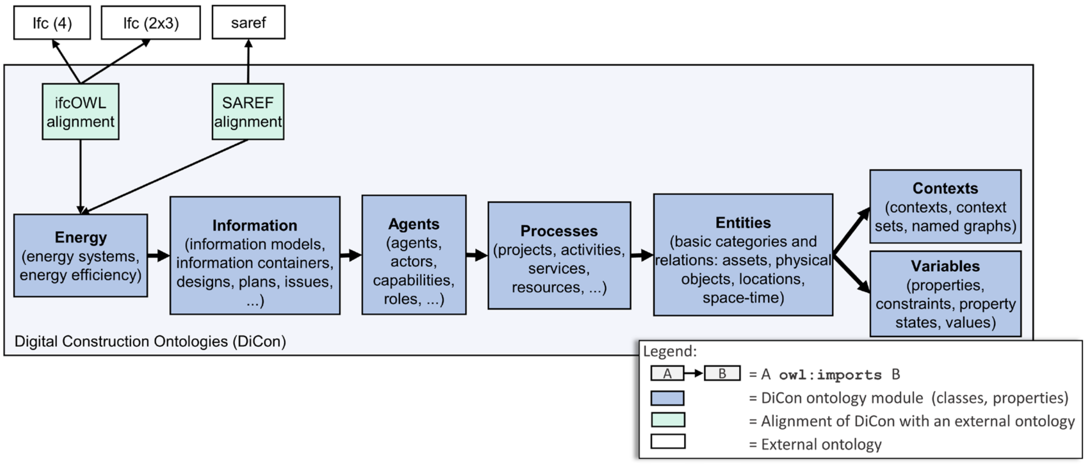

Figure 22: Integration of other ontologies into DICES ontology

6 Ontology Evaluation

6.1 Evaluation Methodologies Used

The developed ontologies in the BIM4EEB project are evaluated using user-based approach. The methods considered for the evaluation are shown in the below tables and these are considered from the state of art analysis. The ontologies are checked against the evaluation methods 1 to 7. Since the developed ontologies are available on GitHub open source repository, one can validate ontologies using methods 1 to 7.

Table 19: Ontology evaluation approach considered for BIM4EEB ontologies

OEP	OEC	OEM								
		1	2	3	4	5	6	7	11	19
Correctness	Accuracy			●						●
	Completeness			●			●	●	●	●
	Conciseness					●				
	Consistency			●	●	●				
Quality	Adaptability						●			●
	Clarity	●	●	●	●		●	●		
	Computational Efficiency						●	●		
OEAs		Vocabulary							Syntax	Context
OE Ap	User Based	i							i	i

Secondly, the syntax is validated using the ontology method validate against the XML schema. For this OWL validator[20] tool is used and it checked the ontologies using the OWL 2 profile. The validation of ontologies according to context is performed by using the competency questions (CQs). In this document, we used CQs are the primary thing to evaluate the ontologies.

6.1.1 Competency Questions

Semantic web development has been increased in the last decade. Ontology development plays one of the significant roles in the semantic web. To support ontologies, several methodologies and tools have appeared in the life cycle phases of the ontology. Ontology requirements define as competency questions (CQ's) [Uschold,1996] in the several methodologies used for ontology development (das Almas--Bahia-Brazil, n.d.). CQ's serve as functional requirements in the sense that the developed ontology or an

[20] http://visualdataweb.de/validator/

ontology-based information system should be able to answer them [Wisniewski,2018].

“Competency Questions (CQs) are natural language questions outlining and constraining the scope of knowledge represented by an ontology” [Wisniewski,2018].

The concept of these competency questions is adopted in the harmonization process. Several different competency questions are developed in the other deliverables of WP3. These competency questions are categorized in this document according to the ontologies developed in the BIM4EEB project. Along with this, the following recommendations are taken into consideration in developing the CQ’s:

- Avoid Redundancy in the questions;
- Avoid incomplete sentences that cannot be properly understood;
- Avoid sentences that are not CQs;
- Avoid questions beyond the expressive power of a DL-based ontology language [Ren,2014].

In the evaluation criteria, a simple knowledge base is created for the ontologies and SPARQL queries are framed for each competency question and queried against the data. The result of the query is checked with the expected results.

6.2 Evaluation of Ontologies

The evaluation of ontologies ic carried out by using sample data produced for the DiCon ontologies. The annex III contains sample data for eac ontology. In this section, SPARQL queries are developed for each competency question and queried against the sample data. Later, query execution time is recorded for the each query and listed in this section.

The prefixes used for the data and SPARQL query is listed in annex III under each ontology to avoid any duplication in the document.

6.2.1 Evaluation of Digital Construction Contexts (DICC) ontology

dicc-cq1 How to store and manage datasets separately? (for versions or alternatives)

SPARQL Query:

```
SELECT DISTINCT ?p ?o
WHERE
 {
{ :Renovation1 ?p ?o }  UNION  { GRAPH ?g { :Renovation1 ?p ?o }}
}
```

Query Result returned:

Table 20: SPARQL query result for dicc-cq1

p	o
rdf:type	dicp:RenovationProject
dicp:hasObject	:Building
dicp:hasSubActivity	:R1Design
dicp:hasSubActivity	:R1Construction
dicp:hasSubActivity	:R1Handover
dicp:hasSubActivity	:R1Procurement

Query Execution Time: <1 ms

There is additional content in the result from context :R1MP2. The context can be managed independently by deleting and insert content.

SPARQL Query is repeated:

SELECT DISTINCT ?p ?o
WHERE { { :Renovation1 ?p ?o } UNION { GRAPH ?g { :Renovation1 ?p ?o }}}

Query Result returned:

Table 21: SPARQL query result for dicc-cq1

p	o
rdf:type	dicp:RenovationProject
dicp:hasObject	:Building
dicp:hasSubActivity	:R1Design
dicp:hasSubActivity	:R1Construction
dicp:hasSubActivity	:R1Handover

The content created by the additional context :R1MP2 has disappeared, and the system is in the state preceding its addition.

Query Execution Time: <1 ms

dicc-cq2 Is the given statement true in the given context?

SPARQL Query:

An example query whether the :Apartment102 is adjacent to :Apartment101 according to the context :R1 can be made as follows:

```
ASK
 {
:R1 dicc:hasContent ?g .
    GRAPH ?g { :Apartment102 dice:hasAdjacentElement :Apartment101 }
}
```

Query Result returned:

Yes.

Query Execution Time: <1 ms

dicc-cq3 What statements hold (are true) in the given context?

SPARQL Query:

The following query gives all statements that hold in the current state defined by :DefaultContextSet:

```
CONSTRUCT { ?s ?p ?o }
WHERE
{
 :DefaultContextSet dicc:hasContent ?g .
     GRAPH ?g { ?s ?p ?o . }
}
```

Query Returned result:

```
@prefix : <http://example.com/id/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
:Renovation1 a <https://w3id.org/digitalconstruction/0.5/Processes#RenovationProject>;
<https://w3id.org/digitalconstruction/0.5/Processes#hasObject> :Building .
```

Query Execution Time: <1 ms

dicc-cq4 What is the difference between a dataset in two different contexts?(e.g., between versions)

SPARQL Query:

The following query computes the difference between two contexts, :R1MP1 and :R1MP2, first in forward direction (what additional statements are in :R1MP1):

```
SELECT ?s ?p ?o
WHERE { :R1MP1 dicc:hasContent ?g1 .
      :R1MP2 dicc:hasContent ?g2 .
      GRAPH ?g1 { ?s ?p ?o }
      FILTER NOT EXISTS { GRAPH ?g2 { ?s ?p ?o } } }
```

Query Returned result:

Table 22: SPARQL query result for dicc-cq4

s	p	o
:R1Design	time:overlaps	:R1Construction

Then the difference in the backward direction (what additional statements are in :R1MP2):

Query Execution Time: <1 ms

SPARQL Query:

```
PREFIX : <http://example.com/id/>
PREFIX dicc: <https://w3id.org/digitalconstruction/0.5/Contexts#>
SELECT ?s ?p ?o
WHERE { :R1MP2 dicc:hasContent ?g1 .
      :R1MP1 dicc:hasContent ?g2 .
      GRAPH ?g1 { ?s ?p ?o }
      FILTER NOT EXISTS { GRAPH ?g2 { ?s ?p ?o } } }
```

Query Result returned:

s	p	o
:R1Procurement	rdf:type	dicp:Activity
:R1ReplaceWindows	rdf:type	dicp:Activity
:R1InstallHeatPump	rdf:type	dicp:Activity
:Renovation1	dicp:hasSubActivity	:R1Procurement
:R1Construction	dicp:hasSubActivity	:R1ReplaceWindows
:R1Construction	dicp:hasSubActivity	:R1InstallHeatPump
:R1Design	time:overlaps	:R1Procurement
:R1Procurement	time:overlaps	:R1Construction
:R1ReplaceWindows	time:overlaps	:R1InstallHeatPump

Query Execution Time: <1 ms

6.2.2 Evaluation of Digital Construction Variables (DICV) ontology

dicv-cq1 What are all the values of a property of an entity over time? (evolution of the value)

dicv-cq2 What is the quantity kind and unit of a quantitative property?

SPARQL Query:

```
SELECT ?value ?unit ?kind ?time
WHERE
{
  ?s dicv:hasProperty/dicv:hasPropertyState ?ps .
  ?ps dicv:hasValue ?value ;
     dicv:hasTimeOfCreation ?time .
  OPTIONAL { ?ps dicv:hasUnit ?unit } .
  OPTIONAL { ?ps dicv:hasQuantityKind ?kind }
} ORDER BY ?time
```

Query Returned results:

Table 23: SPARQL query result for dicv-cq1 and 2

value	unit	kind	time
800.0	unit:Euro	quantitykind:currency	2021-05-15T12:00:00
840.0	unit:Euro	quantitykind:currency	2021-05-20T12:00:00
860.0	unit:Euro		2021-05-25T12:00:00
860.0	unit:Euro		2021-05-25T12:00:00

Query Execution Time: <1 ms

dicv-cq3 What are the constraints between properties? (e.g., less than, equal)

SPARQL Query:

```
SELECT DISTINCT ?property ?constraint ?comparison
WHERE
{
  {?constraint dicv:constrainsProperty1 ?property } union {?constraint dicv:constrainsProperty2 ?property }
  OPTIONAL { ?constraints dicv:hasComparison ?comparison }
}
```

Query Result returned:

Table 24: SPARQL query result for dicv-cq3

property	constraint	comparison
:AStart	:C1	dicv:Less
:AEnd	:C1	dicv:Less
:AEnd	:C2	dicv:Less
:BStart	:C2	dicv:Less
:BStart	:C3	dicv:Less
:BEnd	:C3	dicv:Less

Query Execution Time: <1 ms

6.2.3 Evaluation of Digital Construction Entities (DICE) ontology

dice-cq1 What entity has a given identifier in the given scope? (e.g., the room number in a building)

SPARQL Query:

```
SELECT ?entity
WHERE {
  ?entity dice:isIdentifiedBy ?identifier .
  ?identifier dice:hasLabel "012345.67890.10479832"  }
```

Query Result returned:

Table 25: SPARQL query result for dice-cq1

entity
http://example.com/id/Device1

Query Execution Time: <1 ms

SPARQL Query:

```
SELECT ?entity
WHERE {
  ?entity dice:isIdentifiedBy ?identifier .
  ?identifier dice:hasLabel "101" ;
          dice:hasScope/dice:hasEntitiesFrom :Building1 . }
```

Result returned:

Table 26: SPARQL query result for dice-cq1

entity
http://example.com/id/Space1

Query Execution Time: <1 ms

dice-cq2 What entities are classified in a given category in the given scope? (e.g., EG000819 in ETIM)

SPARQL Query:

```
SELECT ?entity
WHERE
{
  ?entity dice:isClassifiedBy ?category .
       ?category dice:hasScope :GTIN ;
              dice:hasLabel "012345.67890" .
}
```

Query Result returned:

Table 27: SPARQL query result for dice-cq2

entity
http://example.com/id/Device1

Query Execution Time: <1 ms

SPARQL Query:

```
SELECT ?entity
WHERE
{
  ?entity dice:isClassifiedBy ?category .
  ?category dice:hasScope :ETIM ;
          dice:hasLabel "EC000819" .
}
```

Query Result returned:

Table 28: SPARQL query result for dice-cq2

entity
http://example.com/id/Device1
http://example.com/id/Device2

Query Execution Time: <1 ms

dice-cq3 What identifiers/categories does an entity have and in which scopes?

SPARQL Query:

```
SELECT ?entity ?type ?label ?scope
WHERE
{
  {?entity dice:isClassifiedBy ?labeling } union { ?entity dice:isIdentifiedBy ?labeling}
  ?labeling a ?type .
   FILTER ((?type = dice:Identifier) || (?type =  dice:Category))
  ?labeling dice:hasLabel ?label ;
          dice:hasScope ?scope .
}
```

Query Result returned:

Table 29: SPARQL query result for dice-cq3

entity	type	label	scope
http://example.com/id/Device1	https://w3id.org/digitalconstruction/0.5/Entities#Identifier	012345.67890.10479832	http://example.com/id/SGTIN96
http://example.com/id/Space1	https://w3id.org/digitalconstruction/0.5/Entities#Identifier	101	http://example.com/id/B1Rooms
http://example.com/id/Device1	https://w3id.org/digitalconstruction/0.5/Entities#Category	EC000819	http://example.com/id/ETIM
http://example.com/id/Device1	https://w3id.org/digitalconstruction/0.5/Entities#Category	012345.67890	http://example.com/id/GTIN
http://example.com/id/Device2	https://w3id.org/digitalconstruction/0.5/Entities#Category	EC000819	http://example.com/id/ETIM

Query Execution Time: <1 ms

dice-cq4 What type and instance have been assigned to an entity? (prescriptive, as in product selection)

SPARQL Query:

```
SELECT ?entity ?typeLabel ?typeScope ?label ?scope
WHERE
{
  ?entity dice:isAssignedToType ?type .
  ?type dice:hasLabel ?typeLabel ;
       dice:hasScope ?typeScope .
  OPTIONAL { ?entity dice:isAssignedToInstance ?identifier .
    ?identifier dice:hasLabel ?label ;
             dice:hasScope ?scope }
}
```

Query Result returned:

Table 30: SPARQL query result for dice-cq4

entity	typeLabel	typeScope	label	scope
:P1	9888001	:ETIM	9888001	:SGTIN96

Query Execution Time: <1 ms

dice-cq5 What entities are positioned in the given location?

SPARQL Query:

```
SELECT ?entity ?loc
WHERE
{ ?entity dice:isLocatedIn ?loc }
```

Query Result returned:

Table 31: SPARQL query result for dice-cq5

entity	loc
http://example.com/id/Sensor1	http://example.com/id/Loc1
http://example.com/id/Agent1	http://example.com/id/Loc1

Query Execution Time: <1 ms

dice-cq6 What are the parts of a building object?

SPARQL Query:

```
SELECT ?whole ?part
WHERE
```

{ ?whole dice:hasContinuantPart ?part }

Query Result returned:

Table 32: SPARQL query result for dice-cq6

whole	part
http://example.com/id/BO1	http://example.com/id/B02
http://example.com/id/BO1	http://example.com/id/B03
http://example.com/id/BO1	http://example.com/id/B04
http://example.com/id/B02	http://example.com/id/B021
http://example.com/id/B02	http://example.com/id/B022
http://example.com/id/B03	http://example.com/id/B031
http://example.com/id/B03	http://example.com/id/B032
http://example.com/id/B03	http://example.com/id/B033

Query Execution Time: <1 ms

6.2.4 Evaluation of Digital Construction Processes (DICP) ontology

dicp-cq1 What subactivities (or leaf-level subactivities) does the activity has?

SPARQL Query:

SELECT ?activity ?subactivity
WHERE
{ ?activity dicp:hasSubActivity ?subactivity }

Query Result returned:

Table 33: SPARQL query result for dicp-cq1

activity	subactivity
http://example.com/id/Renovation1	http://example.com/id/R1Design
http://example.com/id/Renovation1	http://example.com/id/R1Procurement
http://example.com/id/Renovation1	http://example.com/id/R1Construction
http://example.com/id/Renovation1	http://example.com/id/R1Handover
http://example.com/id/R1Construction	http://example.com/id/R1ReplaceWindows
http://example.com/id/R1Construction	http://example.com/id/R1InstallHeatPump

Query Execution Time: <1 ms

dicp-cq2 What entities (or input/output entities) is the given activity acting on?

SPARQL Query:

SELECT ?activity ?object
WHERE
{ ?activity dicp:hasObject ?object }

Query Result returned:

Table 34: SPARQL query result for dicp-cq2

activity	object
http://example.com/id/DryWallConstruction1	http://example.com/id/DryWall1
http://example.com/id/EquipmentTransfer1	http://example.com/id/NailGun1
http://example.com/id/EquipmentTransfer1	http://example.com/id/SpiritLevel1
http://example.com/id/EquipmentTransfer1	http://example.com/id/Drill1
http://example.com/id/EquipmentTransfer1	http://example.com/id/SurfaceGrinder1
http://example.com/id/FrameErection1	http://example.com/id/DryWall1
http://example.com/id/BackboardInstallation1	http://example.com/id/DryWall1

http://example.com/id/ElectricalWiring1	http://example.com/id/DryWall1
http://example.com/id/FrontboardInstallation1	http://example.com/id/DryWall1
http://example.com/id/WallLeveling1	http://example.com/id/DryWall1
http://example.com/id/WallPainting1	http://example.com/id/DryWall1

Query Execution Time: <1 ms

SPARQL Query:

SELECT ?activity ?object
WHERE
{ ?activity dicp:hasOutputObject ?object }

Query Result returned:

Table 35: SPARQL query result for dicp-cq2

activity	object
http://example.com/id/DryWallConstruction1	http://example.com/id/DryWall1
http://example.com/id/FrameErection1	http://example.com/id/DryWall1

Query Execution Time: <1 ms

dicp-cq3 In what location (or initial/final location) is the activity taking place?

SPARQL Query:

SELECT ?activity ?location
WHERE
{ ?activity dicp:hasLocation ?location }

Query Result returned:

Table 36: SPARQL query result for dicp-cq3

activity	location
http://example.com/id/DryWallConstruction1	http://example.com/id/Apartment101
http://example.com/id/FrameErection1	http://example.com/id/Apartment101
http://example.com/id/BackboardInstallation1	http://example.com/id/Apartment101
http://example.com/id/ElectricalWiring1	http://example.com/id/Apartment101
http://example.com/id/FrontboardInstallation1	http://example.com/id/Apartment101
http://example.com/id/WallPainting1	http://example.com/id/Apartment101
http://example.com/id/WallLeveling1	http://example.com/id/Apartment101

Query Execution Time: <1 ms

dicp-cq4 What equipment are needed in the execution of an activity?

SPARQL Query:

SELECT ?activity ?type ?capability
WHERE
{
?activity dicp:hasResourceRequirement ?requirement .
?requirement dicp:requiresResourceType ?type ;
dicp:requiresResourceCapability ?capability . }

QUERY Result returned:

Table 37: SPARQL query result for dicp-cq4

activity	type	capability
:FrameErection1	:NailingTool	:AutomaticNailingCapability
:ElectricalWiring1	:DrillingTool	:WoodDrillingCapability

Query Execution Time: <1 ms

dicp-cq5 What resources have been assigned for the execution of an activity?

SPARQL Query:

SELECT ?activity ?resource
WHERE
{ ?activity dicp:hasAssignedResource/dicp:isRoleOf ?resource }

Query Result returned:

Table 38: SPARQL query result for dicp-cq5

activity	resource
http://example.com/id/ElectricalWiring1	http://example.com/id/Drill1

Query Execution Time: <1 ms

dicp-cq6 What is the time when the activity is executed? (planned and actual times)

SPARQL Query:

```
SELECT ?activity ?plannedStart ?plannedEnd ?actualStart ?actualEnd
WHERE
{
  ?context dicc:hasContent ?g .
  GRAPH ?g {
    ?activity dice:occupiesTimeInterval ?plannedInterval .
    ?plannedInterval dice:hasStart/time:inXSDDateTime ?plannedStart ;
                  dice:hasEnd/time:inXSDDateTime ?plannedEnd }
    ?activity dice:occupiesTimeInterval ?actualInterval .
    ?actualInterval dice:hasStart/time:inXSDDateTime ?actualStart ;
                 dice:hasEnd/time:inXSDDateTime ?actualEnd }
```

Query Result returned:

Table 39: SPARQL query result for dicp-cq6

activity	plannedStart	plannedEnd	actualStart	actualEnd
:FrameErection1	2021-05-15 T08:00:00	2021-05-15 T16:00:00	2021-05-16 T09:00:00	2021-05-17 T12:00:00

Query Execution Time: <1 ms

6.2.5 Evaluation of Digital Construction Agents (DICA) ontology

dica-cq1 Who is the agent of an activity? (a person or organization)

SPARQL Query:

SELECT ?activity ?agent
WHERE
{ ?activity dica:hasAgent ?agent }

Query Result returned:

Table 40: SPARQL query result for dica-cq1

activity	agent
:FrameErection1	:Carpenter1
:BackboardInstallation1	:Carpenter1
:ElectricalWiring1	:Electrician1
:FrontboardInstallation1	:Carpenter1
:WallLeveling	:Painter1
:WallPainting	:Painter1

Query Execution Time: <1 ms

dica-cq2 What is the consortium of the given renovation project?

SPARQL Query:

```
SELECT ?project ?party ?appointed
WHERE
{
  ?project dica:hasExecutingAgent ?team .
  ?team dice:hasMember ?party, ?appointed .
  ?appointment a dica:Appointment ;
            dica:hasAppointingParty ?party ;
            dica:hasAppointedParty ?appointed }
```

Query Result returned:

Table 41: SPARQL query result for dica-cq2

project	party	appointed
:Project1	:ConstructionManager1	:Architect1
:Project1	:ConstructionManager1	:Contractor1

Query Execution Time: <1 ms

dica-cq3 Who is leading the consortium of the given renovation project?

SPARQL Query:

```
SELECT ?project ?lead
WHERE
{ ?project dica:hasExecutingAgent/dica:hasLeadAppointedParty ?lead }
```

Query Result returned:

Table 42: SPARQL query result for dica-cq3

project	lead
:Project1	:ConstructionManager1

Query Execution Time: <1 ms

dica-cq4 Who are the stakeholders (owners and occupants) related to an activity?

SPARQL Query:

```
SELECT ?activity ?owner ?occupant
WHERE {
  ?activity dicp:hasLocation ?loc .
  ?loc a dice:ResidentialUnit .
  ?owner dice:isOwnerOf ?loc .
  OPTIONAL { ?occupant dice:isOccupantIn ?loc } }
```

Query Result returned:

Table 43: SPARQL query result for dica-cq4

activity	owner	occupant
:DryWallConstruction1	:ResidentialOwner1	:Occupant1
:FrameErection1	:ResidentialOwner1	:Occupant1
:BackboardInstallation1	:ResidentialOwner1	:Occupant1
:ElectricalWiring1	:ResidentialOwner1	:Occupant1
:FrontboardInstallation1	:ResidentialOwner1	:Occupant1
:WallLeveling1	:ResidentialOwner1	:Occupant1
:WallPainting1	:ResidentialOwner1	:Occupant1
:Activity1	:ResidentialOwner1	:Occupant1

Query Execution Time: <1 ms

6.2.6 Evaluation of Digital Construction Information (DICI) ontology

dici-cq1 What is the information model of the given renovation project? (according to ISO 19650)

SPARQL Query:

SELECT ?project ?pim
WHERE
{ ?pim a dici:ProjectInformationModel ;
dici:isAbout ?project }

Query Result returned:

Table 44: SPARQL query result for dici-cq1

project	pim
http://example.com/id/Project1	http://example.com/id/PIM1

Query Execution Time: <1 ms

dici-cq2 What information containers are active in the current state? (contain current information)

SPARQL Query:

SELECT ?project ?container
WHERE
{
?pim dici:isAbout ?project ;
dici:hasDefaultContainerSet ?defaultSet .
?defaultSet dicc:hasActiveContext ?container }

Query Result returned:

Table 45: SPARQL query result for dici-cq2

project	container
:Project1	:ProjectExecution
:Project1	:ArchLOD350
:Project1	:StructLOD350

Query Execution Time: <1 ms

dici-cq3 What information (or output information) does the given activity act on? (information flow)

SPARQL Query:

SELECT ?activity ?information ?input ?output
WHERE
{
?activity dici:hasInformation ?information ;
dici:hasInputInformation ?input ;
dici:hasOutputInformation ?output }

Query Result returned:

Table 46: SPARQL query result for dici-cq3

activity	information	input	output
:QuantityEstimation1	:QuantityEstimationGuide	:ArchBIM1	:QuantityTakeOff1
:QuantityEstimation1	:QuantityEstimationGuide	:StructBIM1	:QuantityTakeOff1
:QuantityEstimation1	:QuantityEstimationGuide	:MEPBIM1	:QuantityTakeOff1

Query Execution Time: <1 ms

dici-cq4 Who produced the specific information content and when? (metadata)

SPARQL Query:

SELECT ?information ?agent ?time
WHERE
{ ?information dici:isCreatedBy ?agent ;
dici:isCreatedAt ?time }

QUERY Result returned:

Table 47: SPARQL query result for dici-cq4

information	agent	time
:QuantityEstimation1	:ProjectManager1	2021-05-15T12:00:00

Query Execution Time: <1 ms

6.2.7 Evaluation of Digital Construction Materials (DICM) ontology

dicm-cq1 How the material object structure is defined?

SPARQL Query:

SELECT ?wallinstance ?layersetinstance
WHERE
{
?wallinstance a dice:BuildingObject .
?wallinstance dicm:hasLayerSet ?layersetinstance .
}

Query Returned Result:

Table 48: SPARQL query result for dicm-cq1

wallinstance	layersetinstance
inst:Wall1	inst:Layerset1

Query Execution Time: <1 ms

dicm-cq2 What are the layers of an BuildingObject ?

SPARQL Query:

SELECT ?wallinstance ?layersetinstance ?layerinstance
WHERE
{
?wallinstance a dice:BuildingObject .
?wallinstance dicm:hasLayerSet ?layersetinstance .
?layersetinstance dicm:hasLayer ?layerinstance .
}

Query Returned Result:

Table 49: SPARQL query result for dicm-cq2

wallinstance	layersetinstance	layerinstance
inst:Wall1	inst:Layerset1	Inst:Layer1
inst:Wall1	inst:Layerset1	Inst:Layer2
inst:Wall1	inst:Layerset1	Inst:Layer3

Query Execution Time: <1 ms

dicm-cq3 What is the adjacent layer of a layer ?

SPARQL query:

SELECT ?layerinstance ?adjacentlayer
WHERE

{
?layerinstance dicm:hasAdjacentLayer ?adjacentlayer
}

Query Returned Result:

Table 50: SPARQL query result for dicm-cq3

layerinstance	adjacentlayer
Inst:Layer1	Inst:Layer2
Inst:Layer2	Inst:Layer3
Inst:Layer2	Inst:Layer1
Inst:Layer3	Inst:Layer2

Query Execution Time: <1 ms

dicm-cq4 What is the material of the layer or building object?

SPARQL Query:

SELECT ?layerinstance ?materialinstance
WHERE
{
?layerinstance dicm:hasMaterial ?materialinstance .
}

Query Returned Result:

Table 51: SPARQL query result for dicm-cq4

layerinstance	materialinstance
Inst:Layer1	inst:Material1
Inst:Layer2	inst:Material2
Inst:Layer3	inst:Material3

Query Execution Time: <1 ms

dicm-cq5 How different materials are classified?

SPARQL Query:

SELECT ?materialinstance ?materialtype
WHERE
{
?layerinstance dicm:hasMaterial ?materialinstance .
?materialinstance a ?materialtype .
FILTER (?materialtype != owl:NamedIndividual)
}

Query Returned Result:

Table 52: SPARQL query result for dicm-cq5

materialinstance	materialtype
inst:Material1	dicm:InorganicNonMetallicMaterial
inst:Material2	dicm:CompositeMaterial
inst:Material3	dicm:InorganicNonMetallicMaterial

Query Execution Time: <1 ms

dicm-cq6 How the material properties are defined?

SPARQL Query:

SELECT ?materialinstance ?value
WHERE

{
?materialinstance dicm:hasThermalConductivity ?value .
}

Query Returned Result:

Table 53: SPARQL query result for dicm-cq6

materialinstance	materialtype
inst:Material1	0.65
inst:Material2	0.54
inst:Material3	0.51

Query Execution Time: <1 ms

dicm-cq7 How the material properties are objectified?

SPARQL Query:

SELECT ?materialinstance ?Property ?value ?unit
WHERE
{
?materialinstance dicv:hasProperty ?Property .
?Property dicv:hasUnit ?unit .
?Property dicv:hasValue ?value .
}

Query Returned Result:

Table 54: SPARQL query result for dicm-cq7

Property	value	unit
inst:property1	0.65	dicu:W_PER_m-K
inst:property2	0.54	dicu:W_PER_m-K
inst:property3	0.51	dicu:W_PER_m-K

Query Execution Time: <1 ms

6.2.8 Evaluation of Digital Construction Occupancy (DICOB) ontology

dicob-cq1 What is the number of occupants in the apartment?

SPARQL Query:

SELECT ?apartment (COUNT(?occupant) as ?occupants)
WHERE
{
?occupant dica:isOccupantIn ?apartment .
} GROUP BY ?apartment

Query Result returned:

Table 55: SPARQL query result for dicob-cq1

apartment	occupants
http://example.com/id/Apartment101	2
http://example.com/id/Apartment102	1

Query Execution Time: <1 ms

dicob-cq2 What are the occupancy schedules performed by the occupants in the building environment?

SPARQL Query:

SELECT ?activity ?person ?start ?end

WHERE
{
?activity a dicob:OccupancyActivity ;
dica:hasAgent ?person ;
dice:occupiesTimeInterval ?interval .
?interval dice:hasStart/time:inXSDDateTime ?start ;
dice:hasEnd/time:inXSDDateTime ?end .
} ORDER BY ?start

Query Result returned:

Table 56: SPARQL query result for dicob-cq2

apartment	activity	person	start	end
:Apartment101	:Cooking1	:Mary	2021-05-15T12:00:00	2021-05-15T12:40:00
:Apartment101	:Vacuuming1	:Mary	2021-05-15T15:00:00	2021-05-15T15:30:00
:Apartment101	:PianoPlaying	:Joe	2021-05-15T18:00:00	2021-05-15T20:00:00

Query Execution Time: <1 ms

dicob-cq3 What is the birthyear of the occupants placed in a residential apartment?

SPARQL Query:

SELECT ?apartment ?person ?year
WHERE
{
?person dica:isOccupantIn ?apartment ;
dica:hasBirthYear ?year }

Query Result returned:

Table 57: SPARQL query result for dicob-cq3

apartment	person	year
:Apartment101	:Mary	1992
:Apartment101	:Joe	1982
:Apartment102	:Bill	1972

Query Execution Time: <1 ms

dicob-cq4 and cq5 What are the minimum/maximum indoor environmental quality (temperature, luminance, noise) in the apartment?

SPARQL Query:

SELECT (MAX(?value) AS ?maxTemp) (MIN(?value) AS ?minTemp) (AVG(?value) AS ?avgTemp)
WHERE
{
?apartment dicv:hasProperty ?property .
?property a dicob:Temperature ;
dicv:hasPropertyState/dicv:hasValue ?value .
} GROUP BY ?property

Query Result returned:

Table 58: SPARQL query result for dicob-c.q4, 5

maxTemp	minTemp	avgTemp
22	18	20.0
22	18	20.0

Query Execution Time: <1 ms

dicob-cq6, cq7, cq8 Which is the temperature/noise/luminance sensor in building zone?

SPARQL Query:

```
SELECT ?sensor ?sensorType ?location
WHERE
{
  ?sensor a ?sensorType ;
        dice:isLocatedIn ?location .
  FILTER NOT EXISTS {
    ?subtype rdfs:subClassOf ?sensorType ;
    FILTER ( ?subtype != ?sensorType ) } }
```

Query Result returned:

Table 59: SPARQL query result for dicob-c.q6, 7, and 8

sensor	sensorType	location
:TemperatureSensor1	dicob:TemperatureSensor	:Apartment101
:NoiseSensor1	dicob:NoiseSensor	:Apartment102
:LuminanceSensor1	dicob:LuminanceSensor	:Apartment102

Query Execution Time: <1 ms

6.2.9 Evaluation of Digital Construction Energy (DICES) ontology

dices-cq1 and cq2 Which is the generation/storage system in building zone?

SPARQL Query:

```
SELECT ?device ?location
WHERE
{
  {?device a dices:Generator } union {?device a dices:BatteryStorage }
        ?device dice:isLocatedIn ?location }
```

Query Result returned:

Table 60: SPARQL query result for dices-c.q1 and 2

device	location
:Generator1	:BuildingUnit1
:Battery1	:Apartment102

Query Execution Time: <1 ms

dices-cq3 and cq4 What is the operational service for the HVAC_1/Photovoltics_1 device?

SPARQL Query:

```
SELECT ?device ?service
WHERE
{
  {?device a dices:HVACDevice } union {?device a dices:Photovotaics }
   ?device dices:offersService ?service }
```

Query Result returned:

Table 61: SPARQL query result for dices-c.q3 and 4

device	service
:HVAC_1	:ComfortManagement1
:HVAC_1	:ServiceAggregation1
:Photovotaics_1	:SelfConsumptionOptimiation1

Query Execution Time: <1 ms

dices-cq5 What are the LCA values of the Photovoltaics_1?

SPARQL Query:

```
SELECT ?lca ?impactType ?impactValue
WHERE
{
 ?lca a dices:LifeCycleAssessment ;
      dici:isAbout ?device ;
      dici:hasLifeCycleImpact ?impact  .
 ?impact a ?impactType ;
        dicv:hasValue ?impactValue .
 FILTER NOT EXISTS {
  ?subtype rdfs:subClassOf ?impactType ;
  FILTER ( ?subtype != ?impactType ) } }
```

Query Result returned:

Table 62: SPARQL query result for dices-c.q5

lca	impactType	impactValue
:LifeCycleAssessment	dices:OzoneDepletionPotential	0.01
:LifeCycleAssessment	dices:HumanToxicityPotential	2.26E-3

Query Execution Time: <1 ms

dices-cq6 What are the LCC values of the Photovoltaics_1?

SPARQL Query:

```
SELECT ?lca ?costType ?costValue
WHERE
{
 ?lca dici:documentsProperty ?lcaProperty  .
 ?lcaProperty a ?costType ;
           dicv:hasValue ?costValue .
 FILTER NOT EXISTS {
  ?subtype rdfs:subClassOf ?costType ;
  FILTER ( ?subtype != ?costType ) } }
```

Query Result returned:

Table 63: SPARQL query result for dices-c.q6

lca	costType	costValue
:LifeCycleAssessment	dices:CapitalCost	2000
:LifeCycleAssessment	dices:EngineeringCost	1000
:LifeCycleAssessment	dices:MaintenanceCost	1200

Query Execution Time: <1 ms

dices-cq7 Which is the meter system in building zone?

SPARQL Query:

```
SELECT ?meter ?location
WHERE
{
 ?meter a dices:Meter ;
     dice:isLocatedIn ?location }
```

Query Result returned:

Table 64: SPARQL query result for dices-c.q7

meter	location
:Meter1	:Apartment101

Query Execution Time: <1 ms

dices-cq8 What is the primary energy conversion factor for the meter system?

SPARQL Query:

```
SELECT ?meter ?value
WHERE
{
  ?meter dices:hasProperty ?property .
  ?property a dices:PrimaryEnergyFactor ;
         dicv:hasValue ?value }
```

Query Result returned:

Table 65: SPARQL query result for dices-c.q8

meter	value
:Meter1	0.3

Query Execution Time: <1 ms

dices-cq9 What is the CO2 emission conversion factor for the meter system?

SPARQL Query:

```
SELECT ?meter ?property ?value
WHERE
{
  ?meter dices:hasProperty ?property .
  ?property a dices:FossilFuelPotential ;
          dicv:hasValue ?value }
```

Query Result returned:

Table 66: SPARQL query result for dices-c.q9

meter	value
:Meter1	0.4

Query Execution Time: <1 ms

6.2.10 Evaluation of Digital Construction Lifecycle (DICL) ontology

dicl-cq1: How can the BIM data representation be adjusted or modified to different LOD systems?

dicl-cq2: what is the link between the lod system and its levels?

SPARQL Query:

```
SELECT ?system ?levels
WHERE {
?system dicl:hasLevel ?levels.
}ORDER BY ASC(?levels)
```

Query Result returned:

Table 67: SPARQL query result for dicl-c.q1 and c.q2

system	levels
diclvl:USA_BIMForum	diclvl:AsBuilt
diclvl:USA_BIMForum	diclvl:AsDesigned
diclvl:USA_BIMForum	diclvl:LOD_100
diclvl:USA_BIMForum	diclvl:LOD_200
diclvl:USA_BIMForum	diclvl:LOD_300
diclvl:USA_BIMForum	diclvl:LOD_350

diclvl:USA_BIMForum	diclvl:LOD_400
diclvl:USA_BIMForum	diclvl:LOD_500

Query Execution Time: <1 ms

dicl-cq3: what is the relation between the lod classification-level and the lod scale?

SPARQL Query:

SELECT ?System ?Sub_type ?Scale
Where{
?System dicl:hasLevel ?Sub_type.
?Sub_type dicl:hasSubLevel ?Scale .
}ORDER BY ASC(?Scale)

Query Result returned:

Table 68: SPARQL query result for dicl-c.q3

System	Sub_type	Scale
diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_100
diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_200
diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_300
diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_350
diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_400
diclvl:USA_BIMForum	diclvl:AsBuilt	diclvl:LOD_500

Query Execution Time: <1 ms

dicl-cq4: what is the relation between lod scales?

SPARQL Query:

SELECT ?level ?Nextlevels
WHERE {
?system dicl:hasLevel ?level.
optional{?level dicl:hasNextLevel ?Nextlevels.}
}ORDER BY ASC(?level)

Query Result returned:

Table 69: SPARQL query result for dicl-c.q4

level	Nextlevels
diclvl:AsBuilt	
diclvl:AsDesigned	diclvl:AsBuilt
diclvl:LOD_100	diclvl:LOD_500
diclvl:LOD_100	diclvl:LOD_300
diclvl:LOD_100	diclvl:LOD_400
diclvl:LOD_100	diclvl:LOD_350
diclvl:LOD_100	diclvl:LOD_200
diclvl:LOD_200	diclvl:LOD_500
diclvl:LOD_200	diclvl:LOD_300
diclvl:LOD_200	diclvl:LOD_400
diclvl:LOD_200	diclvl:LOD_350
diclvl:LOD_300	diclvl:LOD_500
diclvl:LOD_300	diclvl:LOD_400
diclvl:LOD_300	diclvl:LOD_350
diclvl:LOD_350	diclvl:LOD_500
diclvl:LOD_350	diclvl:LOD_400

diclvl:LOD_400	diclvl:LOD_500
diclvl:LOD_500	

Query Execution Time: <1 ms

dicl-cq5: how is the bim object is represented?

SPARQL Query:

Select Distinct ?BuildingObject ?ObjectName
where {
?BuildingObject a dice:BuildingObject .
?BuildingObject rdfs:label ?ObjectName .
}

Query Result returned:

Table 70: SPARQL query result for dicl-c.q5

BuildingObject	ObjectName
:GlobalId_2af9a9d4-6443-4ac2-b74d-e584c11f4652	"Basic Wall:500+100:2145690"

Query Execution Time: <1 ms

dicl-cq6: how to represent multiple versions of information about the same object?

SPARQL Query:

Select Distinct ?ObjectName ?PropertyName ?PropertyState ?PropertyValue
where {
?BuildingObject dicv:hasProperty/dicv:hasPropertyState ?PropertyState .
?BuildingObject rdfs:label ?ObjectName .
?PropertyState dicl:hasPropertyName ?PropertyName .
?PropertyState dicv:hasValue ?PropertyValue.
}

Query Result returned:

Table 71: SPARQL query result for dicl-c.q6

ObjectName	PropertyName	PropertyState	PropertyValue
"Basic Wall:500+100:2145690"	"ThermalTransmittance"	:IfcPropertySingleValue_2024_LOD300	"1.087"^^xsd:double
"Basic Wall:500+100:2145690"	"ThermalTransmittance"	:IfcPropertySingleValue_2024_LOD200	"1.717"^^xsd:double

Query Execution Time: <1 ms

dicl-cq7: how the object properties and values for a specific lod level are defined?

SPARQL Query:

Select Distinct ?ObjectName ?PropertyName ?PropertyValue ?LODLevel
where {
?BuildingObject dicv:hasProperty/dicv:hasPropertyState ?PropertyState .
?BuildingObject rdfs:label ?ObjectName .
?PropertyState dicl:hasPropertyName ?PropertyName .
{?PropertyState dicv:hasValue ?PropertyValue;
dicl:hasLODLevel ?Level.}
?Level rdfs:label ?LODLevel. }

Query Result returned:

Table 72: SPARQL query result for dicl-c.q7

ObjectName	PropertyName	PropertyValue	LODLevel
"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"1.087"^^xsd:double	"LOD 300"
"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"1.717"^^xsd:double	"LOD 200"

Query Execution Time: <1 ms

dicl-cq8: what are the sources for lod data?

SPARQL Query:

```
Select Distinct ?ObjectName ?PropertyName ?PropertyValue ?LODLevel ?Source
where {
        ?BuildingObject dicv:hasProperty/dicv:hasPropertyState ?PropertyState .
        ?BuildingObject rdfs:label ?ObjectName .
        ?PropertyState dicl:hasPropertyName ?PropertyName .
        {?PropertyState dicv:hasValue ?PropertyValue;
                    dicl:hasLODLevel ?Level;
                    dicl:isDerivedFrom ?Source}
       ?Level rdfs:label ?LODLevel.        }
```

Query Result returned:

Table 73: SPARQL query result for dicl-c.q8

ObjectName	PropertyName	PropertyValue	LODLevel	Source
"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"1.087"^^xsd:double	"LOD 300"	:Polish_site_LOD300_model
"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"1.717"^^xsd:double	"LOD 200"	:Polish_site_LOD200_model

Query Execution Time: <1 ms

dicl-cq9: how the activities are defined in the renovation workflow?

dicl-cq10: how to identify the sequence of activities?

SPARQL Query:

```
SELECT ?BuildingStage ?Activity
WHERE
{
?BuildingStage dicl:hasActivity ?Activity .
}
```

Query Result returned:

Table 74: SPARQL query result for dicl-c.q9 and 10

BuildingStage	Activity
dicstg:3.5_DetailedDesign	:125_BuildingPermissions
dicstg:3.5_DetailedDesign	:124_TechnicalDetails
dicstg:3.5_DetailedDesign	:123_ProductionOfPlans

Query Execution Time: <1 ms

dicl-cq11: how are the stakeholders or agents related to activities?

SPARQL Query:

SELECT ?Stakeholder ?relation ?Activity
WHERE
{
?BuildingStage dicl:hasActivity ?Activity .
?Stakeholder rdf:type dica:Agent ;
?relation ?Activity .
}

Query Result returned:

Table 75: SPARQL query result for dicl-c.q11

Stakeholder	relation	Activity
:Architectural_Designer	dicl:providesTo	:125_BuildingPermissions
:Building_Services_Designer	dicl:consumesFrom	:123_ProductionOfPlans
:Building_Services_Designer	dicl:providesTo	:125_BuildingPermissions
:Building_Services_Designer	dicl:processFrom	:123_ProductionOfPlans
:Project_Leader	dicl:consumesFrom	:125_BuildingPermissions

Query Execution Time: <1 ms

dicl-cq12: how does the renovation process linked to bim data represented in lod-sensitive manner?

SPARQL Query:

SELECT ?BuildingStage ?Activity ?ObjectName ?PropertyName ?PropertyValue
WHERE
{
?BuildingStage dicl:hasActivity ?Activity .
?Activity dicp:hasObject ?BuildingObject .
?BuildingObject dicv:hasProperty/dicv:hasPropertyState ?PropertyState .
?BuildingObject rdfs:label ?ObjectName .
?PropertyState dicl:hasPropertyName ?PropertyName .
{?PropertyState dicv:hasValue ?PropertyValue;
dicl:hasLODLevel diclvl:LOD_300.}
}

Query Result returned:

Table 76: SPARQL query result for dicl-c.q12

BuildingStage	Activity	ObjectName	PropertyName	PropertyValue
dicstg:3.5_Detailed Design	:123_ProductionOf Plans	"Basic Wall:500+100:214 5690"	"Width"	"600"^^xsd:dou ble
dicstg:3.5_Detailed Design	:123_ProductionOf Plans	"Basic Wall:500+100:214 5690"	"ThermalTransmitt ance"	"1.087"^^xsd:d ouble
dicstg:3.5_Detailed Design	:124_TechnicalDeta ils	"Basic Wall:500+100:214 5690"	"Width"	"600"^^xsd:dou ble
dicstg:3.5_Detailed Design	:124_TechnicalDeta ils	"Basic Wall:500+100:214 5690"	"ThermalTransmitt ance"	"1.087"^^xsd:d ouble
dicstg:3.5_Detailed Design	:125_BuildingPermi ssions	"Basic Wall:500+100:214	"Width"	"600"^^xsd:dou ble

		5690"		
dicstg:3.5_Detailed Design	:125_BuildingPermissions	"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"1.087"^^xsd:double

Query Execution Time: <1 ms

dicl-cq13: how to represent specific use cases within the renovation workflow?

SPARQL Query:

SELECT ?Usecase ?usecasename ?Activity
WHERE
{
?Usecase dicl:hasRepresents ?Activity;
rdfs:label ?usecasename .
}

Query Result returned:

Table 77: SPARQL query result for dicl-c.q13

Usecase	usecasename	Activity
:UsecaseEM9	"Prepare detailed design"	:125_BuildingPermissions
:UsecaseEM9	"Prepare detailed design"	:124_TechnicalDetails
:UsecaseEM9	"Prepare detailed design"	:123_ProductionOfPlans

Query Execution Time: <1 ms

dicl-cq14: how to enable the representation of multiple bls systems and/or stages?

SPARQL Query:

SELECT Distinct ?Framework
WHERE {
?Framework dicl:hasStage ?Stage .
}

Query Result returned:

Table 78: SPARQL query result for dicl-c.q14

Framework
dicstg:ISO_22263
dicstg:BS_EN_16310

Query Execution Time: <1 ms

dicl-cq15: what is the link between the bls system and its respective stages?

SPARQL Query:

SELECT ?Framework ?Stage
WHERE {
?Framework dicl:hasStage ?Stage .
}

Query Result returned:

Table 79: SPARQL query result for dicl-c.q15

Framework	Stage
dicstg:ISO_22263	dicstg:ISO_Design
dicstg:BS_EN_16310	dicstg:0.BS_EN_Initiative
dicstg:BS_EN_16310	dicstg:1.BS_EN_Initiation
dicstg:BS_EN_16310	dicstg:2.BS_EN_Design

dicstg:BS_EN_16310	dicstg:3.BS_EN_Procurement
dicstg:BS_EN_16310	dicstg:4.BS_EN_Construction
dicstg:BS_EN_16310	dicstg:5.BS_EN_Use
dicstg:BS_EN_16310	dicstg:6.BS_EN_End_of_Life

Query Execution Time: <1 ms

dicl-cq16: what is the relation between stages?

dicl-cq17: what is the relation between the main-stage to sub-stage of other main-stage?

SPARQL Query:

SELECT ?Stage ?relation ?OtherStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Stage=dicstg:4.BS_EN_Construction) .
?Stage ?relation ?OtherStages .}

Query Result returned:

Table 80: SPARQL query result for dicl-c.q16 and 17

Stage	relation	OtherStages
dicstg:4.BS_EN_Construction	dicl:hasNextStage	dicstg:5.BS_EN_Use
dicstg:4.BS_EN_Construction	dicl:hasNextStage	dicstg:6.BS_EN_End_of_Life
dicstg:4.BS_EN_Construction	dicl:hasNextStage	dicstg:5.1.BS_EN_Operation
dicstg:4.BS_EN_Construction	dicl:hasNextStage	dicstg:5.2.BS_EN_Maintenance
dicstg:4.BS_EN_Construction	dicl:hasNextStage	dicstg:6.1.BS_EN_Revamping
dicstg:4.BS_EN_Construction	dicl:hasNextStage	dicstg:6.2.BS_EN_Dismantling
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:0.BS_EN_Initiative
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:1.BS_EN_Initiation
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:2.BS_EN_Design
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:3.BS_EN_Procurement
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:0.1.BS_EN_Market_study
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:0.2.BS_EN_Business_case
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:1.1.BS_EN_Project_initiation
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:1.2.BS_EN_Feasibility_study
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:1.3.BS_EN_Project_definition
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:2.1.BS_EN_Conceptual_design
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:2.2.BS_EN_Preliminary_design
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:2.3.BS_EN_Developed_design
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:2.4.BS_EN_Technical_design
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:2.5.BS_EN_Detailed_design
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:3.1.BS_EN_Procurement
dicstg:4.BS_EN_Construction	dicl:hasPreviousStage	dicstg:3.2.BS_EN_Construction_contracting
dicstg:4.BS_EN_Construction	dicl:hasSubStage	dicstg:4.1.BS_EN_Pre_construction
dicstg:4.BS_EN_Construction	dicl:hasSubStage	dicstg:4.2.BS_EN_Construction
dicstg:4.BS_EN_Construction	dicl:hasSubStage	dicstg:4.3.BS_EN_Commissioning
dicstg:4.BS_EN_Construction	dicl:hasSubStage	dicstg:4.4.BS_EN_Handover

Query Execution Time: <1 ms

dicl-cq18: how to represent the mapping between different bls systems?

SPARQL Query:

SELECT ?Stage ?OtherStages

WHERE {
?Framework dicl:hasStage ?Stage .
?Stage dicl:isRelaventWith ?OtherStages .}
Query Result returned:

Table 81: SPARQL query result for dicl-c.q18

Stage	OtherStages
dicstg:2.BS_EN_Design	dicstg:ISO_Design
dicstg:ISO_Design	dicstg:2.BS_EN_Design

Query Execution Time: <1 ms

7 BIM4EEB toolkit and the ontologies usage

The BIM Management system (BIMMS) is a collaborative platform that cover different stages of a renovation process. BIMeaser and BIMcpd were developed to support decision-making and as a BIM-assisted energy refurbishment assessment tool, in the initiation and design phase. The BIMplanner and BIM4Occupants tools are designed for construction and operational stages. The tools BIMMS, BIMeaser, BIMPlanner, BIM4Occupants, and BIMcpd will use ontologies.

7.1 BIM Management System (BIMMS)

The BIMMS has a hybrid triple store database system based on Virtuoso. This database is used for the RDF data of BIMMS resources, BIM models, sensor data, geospatial data, and data generated by the tools. However, the BIMMS system requires graphs that can manage, organize, and differentiate this information into clusters. For this purpose four BIMMS, linked data graphs are created. The first one is the resource graph, which is used to describe all the resources uploaded in the BIMMS. The second one is ifcdata graph, which is used to store all the IFC BIM models parsed and converted in the MySQL database. The third one is linkeddata graph, which is used to store all the IFC BIM models converted into ifcowl and bot ontologies. The last one is iotdata graph, to store the data from the sensor and their measurements. All these graphs use the ontologies listed in Table 82.

Table 82: Ontologies used in BIMMS tool

BIM4EEB Ontologies	Non-BIM4EEB ontologies
Digital Construction Entities (dice)	Container Ontology (ct)
Digital Construction Variables (dicv)	Vocabulary of Interlinked Datasets (VoID)
Digital Construction Processes (dicp)	Ifcowl
Digital Construction Information (dici)	Building Topology Ontology (BOT)
Digital Construction Occupancy (dicob)	

Apart from the BIMMS linked data graphs, BIMMS provides APIs to handle RDF data. The REST API and SPARQL endpoint are the two APIs developed in the BIMMS. Also, BIMMS provides the ontology administration panel to set a list of ontologies used in BIMMS by the user, ontology viewer to view and consult the ontologies. The RDF data can be imported into the BIMMS by using the Import RDF functionality in the BIMMS. A more detailed description of these functions can be learned in WP4 deliverables.

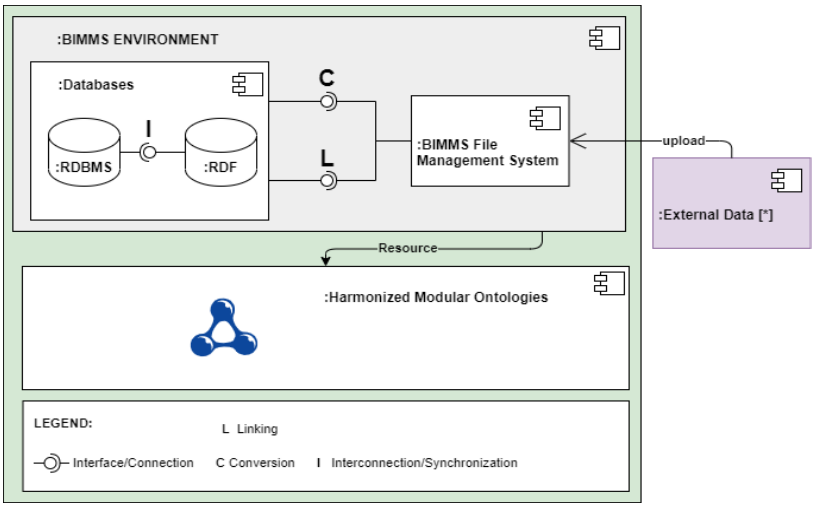

Figure 23: Usage of ontologies in BIMMS

The BIMMS tool implements the linked data principles, represents the data in RDF, and also supports the interfaces called SPARQL Endpoint, REST API. Also, it can access the data using the identifiers, usage of ontologies, and parsing capabilities are included in the tool. The BIMMS uses HTTP URIs for identifiers and supports JSON-LD, TriG, and Turtle data formats. Also, it uses external ontologies.

Table 83: The addressed criteria in BIMMS concerning linked data

System Criteria	**Implementation Details**
Each system that provides data to other systems must:	
P1. Implement the Linked Data Principles (Berners-Lee, 2006) for data sharing	X
P2. Represent the shared data (maybe converted from native data) as RDF graphs with links to additional data	X
P3. Support the specified interfaces	X
P4. Use the specified set of shared ontologies to define the types/properties of entities in shared data	X
Each system that consumes data from other systems must:	
C1. Be able to access the data using specified identifiers, query language, and interfaces	X
C2. Be able to parse the received data and query results	X
C3. Use the specified set of shared ontologies to interpret the types/properties of objects in shared data	X
Identifiers	HTTP URIs, using the HTTPS protocol
Data formats	JSON-LD, TriG and Turtle for rdf data
Interfaces	SPARQL Endpoint REST API
Use of Existing ontologies	X

7.2 BIM4EEB BIMeaser tool

The BIMeaser tool (BIM Early Stage Energy Scenario tool) was developed for the early phase evaluation of residential building refurbishment designs. The BIMeaser tool is able to download BIM models from the BIMMS. Renovation scenarios are loaded for simulation, computing the indicators of building energy performance. These indicators are then compared with reference requirements and the Owner's Project Requirements (OPRs). The target of BIMeaser is to speed up the decision-making process in apartment buildings—a refurbishment project achieved by enhancing the effective informative discussion of different technical design details between experts with different backgrounds in the design team. The aim is to make a better selection of the building's refurbishment design which should meet the building owner's project requirements (OPRs). In addition, the tool should be able to enhance the building's energy performance, cost-effectiveness, and indoor climate conditions for the residents [Daniotti, 2020].

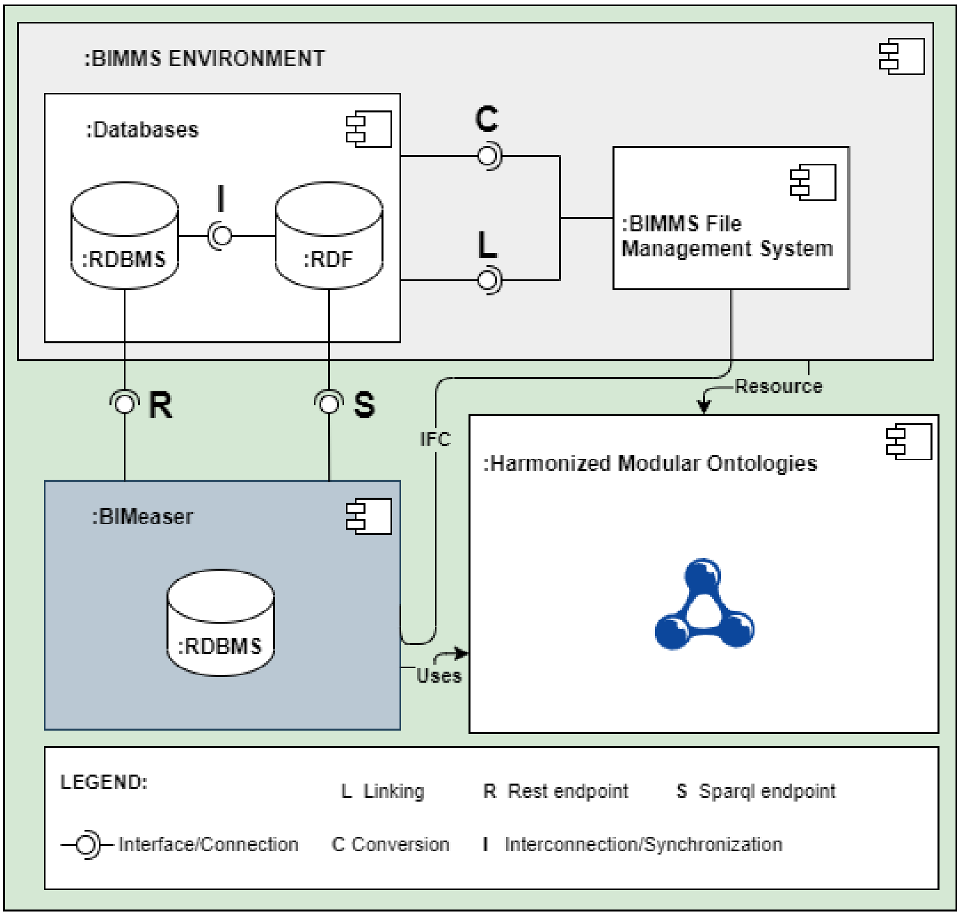

Figure 24: Usage of ontologies in BIMeaser tool

The BIM4EEB modular ontologies are used to establish a relation between the ontologies used to store BIMeaser OPRs. Finally, OPRs calculated with BIMeaser are uploaded into BIMMS in the triple store. Ontologies have been developed and integrated, mapped with the BIMeaser OPRs to develop complete KB. In this process, BIM4EEB ontologies, vocabularies, and the external ontologies listed in Table 84 are used. The ontologies are aligned and imported to their respective aligned modules as shown in the previous sections (ontology harmonization). Therefore, BIMeaser provides an effective collaboration platform for experts with different backgrounds in the design team and can speed up decision-making in building refurbishment projects [Shemeikka,2020].

Table 84: Ontologies used for BIMeaser tool

BIM4EEB Ontologies	Non-BIM4EEB ontologies
Digital Construction Entities (dice)	Building Topology Ontology (bot)
Digital Construction Contexts (dicc)	Quantities, Units, Dimensions and Types (qudt)
Digital Construction Variables (dicv)	QUDT UNITS
Digital Construction Information (dici)	Data Catalog Vocabulary (dcat)
Digital Construction Materials (dicm)	PROV Ontology (prov-o)
Digital Construction Energy (dices)	QUDT Qunatity Kind
Digital Construction Units (dicu)	

More details about the BIMeaser and usage ontologies are described in deliverable D6.6.

The BIMeaser tool provides OPR results in RDF format. The BIMeaser tool supports the criteria P1 to P4 mentioned in the below table. It uses the interfaces SPARQL Endpoint and REST API to share the data with BIMMS. Existing ontologies are used to map data.

Table 85: The addressed criteria in BIMeaser concerning linked data

System Criteria	Implementation Details
Each system that provides data to other systems must:	
P1. Implement the Linked Data Principles (Berners-Lee, 2006) for data sharing	X
P2. Represent the shared data (maybe converted from native data) as RDF graphs with links to additional data	X
P3. Support the specified interfaces	X
P4. Use the specified set of shared ontologies to define the types/properties of entities in shared data	X
Each system that consumes data from other systems must:	
C1. Be able to access the data using specified identifiers, query language, and interfaces	X
C2. Be able to parse the received data and query results	X
C3. Use the specified set of shared ontologies to interpret the types/properties of objects in shared data	X
Identifiers	HTTP URIs, using the HTTPS protocol
Data formats	Turtle for rdf data
Interfaces	SPARQL Endpoint REST API (are part of BIMMS)
Use of Existing ontologies	X

7.3 BIM4EEB BIMPlanner tool

Procurement regulations determine the sequence and scope of information to be shared between stakeholders in an AEC project. The designers (architects and engineers) must ensure that their design proposals do not restrict the choices in the procurement process. Whereas required technical features can be determined (e.g. a minimum U-Value) other properties (e.g. precise geometrical data, connection technologies) can only be specified and included in the design model after the completion of the procurement process. Thus, the design model (and subsequently the ontology-driven repository) will be adjusted and extended with additional specifications. An important part of this adjustment and extension process is to “link” detailed product specifications (from the successful material supplier) to the overall BIM repository.

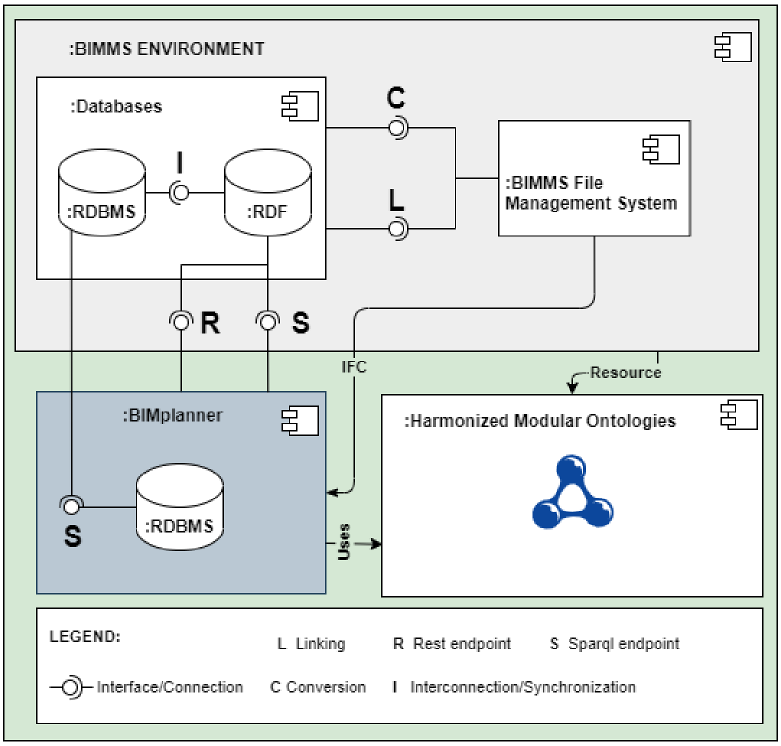

Figure 25: Usage of ontologies in BIMplanner tool

BIMPlanner is a cloud-based planning and management tool for housing renovation projects, currently being developed in the BIM4EEB project. The tool will enhance fast-track construction, improve safety, and reduce disruption to inhabitants with proactive information. The BIMPlanner applies a browser-based user interface. In the backend, data are stored in a graph database in RDF format according to DICon and ifcOWL ontologies. The software architecture also contains a GraphQL-interface between the graph database and BIMPlanner application [Törmä,2020]. Figure 25 illustrates a usage scenario of ontologies in BIMPlanner. More details are described in the WP7 D7.1 and D7.3 deliverables.

Table 86: Ontologies used for BIMplanner tool

BIM4EEB Ontologies and vocabularies	Non-BIM4EEB ontologies
Digital Construction Entities (dice)	Building Topology Ontology (bot)
Digital Construction Contexts (dicc)	Quantities, Units, Dimensions and Types (qudt)
Digital Construction Variables (dicv)	Ontology for Property Management (opm)
Digital Construction Information (dici)	BFO ontology
Digital Construction Processes (dicp)	Wgs84_ps (geo)
Digital Construction Agents (dica)	Ifcowl ontology

BIM4EEB Ontologies and vocabularies	Non-BIM4EEB ontologies
Digital Construction Lifecycle (dicl)	Time ontology in OWL (time)
Digital Construction Materials (dicm)	PROV Ontology (prov)
Digital Construction Occupancy (dicob)	Friend of a Friend (foaf)
Digital Construction Energy (dices)	Organization ontology (org)
Digital Construction Units (dicu)	Real Estate Core ontology (core)
Digital Construction Levels (diclvl)	Container Ontology (ct)
Digital Construction Stages (dicstg)	Semantic Sensor Networks (ssn) Sensor, Observation, Sample and Actuator (sosa)
	Smart Appliances REFerence (saref), SAREF ontology for Buildings (s4bldg)

The BIM4EEB ontologies do not define any fixed design management approach (e.g., what design levels are used or how the design effort is decomposed) nor any fixed processes for procurement management. Instead, the ontologies are designed to be generic and flexible to enable the configuration of desired processes and management levels. The ontologies aim to provide the concepts and properties to represent different approaches and processes that individual projects and companies want to adopt. There are going to be different management approaches depending on many slowly evolving factors, such as regional regulation, the conventions at the geographical area, climate conditions, the legacy of participants, and the level of digitalization. The BIMplanner addresses the criteria defined in the framework. These details are listed in the below table.

Table 87: The addressed criteria in BIMplanner concerning linked data

System Criteria	Implementation Details
Each system that provides data to other systems must:	
P1. Implement the Linked Data Principles (Berners-Lee, 2006)	X
P2. Represent the shared data (maybe converted from native data) as RDF graphs with links to additional data	X
P3. Support the specified interfaces	X
P4. Use the specified set of shared ontologies to define the types/properties of entities in shared data	X
Each system that consumes data from other systems must:	
C1. Be able to access the data using specified identifiers, query language, and interfaces	
C2. Be able to parse the received data and query results	
C3. Use the specified set of shared ontologies to interpret the types/properties of objects in shared data	
Identifiers	HTTP URIs, using the HTTPS protocol
Data formats	JSON-LD, TRiGTurtle for rdf data
Interfaces	SPARQL Endpoint (part of BIMplanner), REST API
Use of Existing ontologies	X

7.4 BIM4EEB BIM4Occupants tool

The main scope of the application is to engage building end-users, (i.e., building occupants) in a collaborative process with other renovation stakeholders, empowering bidirectional communication and enhanced information exchange. In addition, BIM4Occupants stands as the endpoint for building occupants to retrieve information about environmental and energy-related data in the building residential environment. Following the microservices-based concept, different viewpoints on the application are defined by taking into account the features and functionalities as requested during the requirements phase. This modularity of the system is a key element to ensure the customization and easy adaptation of the building occupant renovation process application to the diverse end-user needs. The details about the functionalities served by the different tools are reported in D6.8 and D7.2 respectively.

The scope of this section is to provide a linkage of the development work performed for the BIM4Occupants with the modeling principles as specified in WP3, presented in the following schema

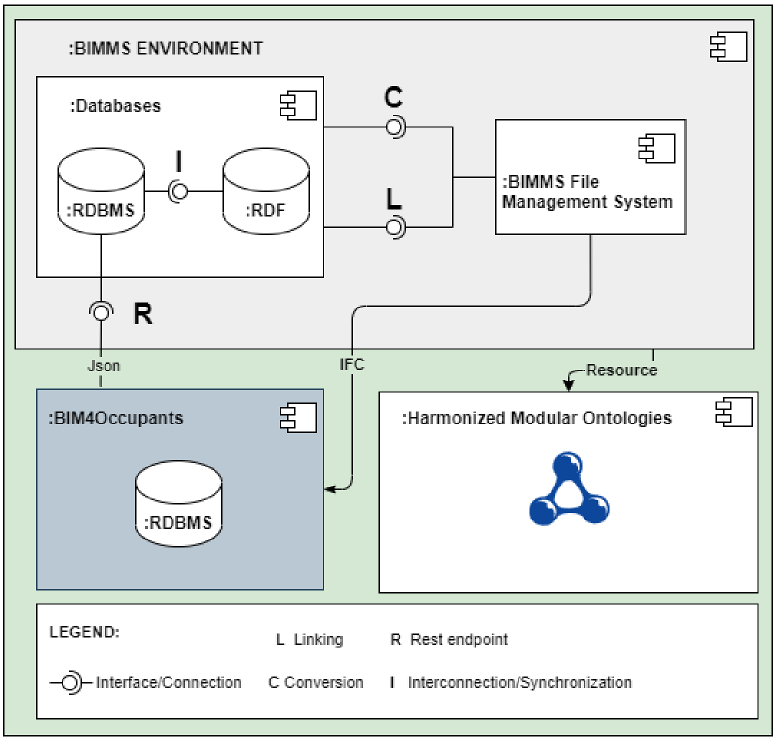

Figure 26: Usage of ontologies for BIM4Occupants

As depicted above, the BIM4Occupants does not incorporate internally any of the ontological models defined in the project, rather it relies on the incorporation of the ontological models in the BIMMS platform to be able to retrieve the different parameters that are specified by the concepts specified in the BIM4EEB linked data framework, expressed in DiCon ontologies. By taking into account the development process of BIM4Occupants, refinement of the different concepts defined in WP3 was performed to be sure that a clear mapping of data needs of the BIM4Occupants applications with the linked data framework of the project exists. More specifically, concepts defined in Occupancy (occupancy profiling and comfort-related parameters) and Energy ontologies (device, measurements, state, etc..) are directly linked with the data needs of the application. In addition, concepts that are defined in DiCon base models (Entities, Variables, Processes, Agents, Information) are also covering the data needs of the BIM4Occupants application. Detailed mapping of the application data needs with the different concepts defined in the BIM4EEB linked data framework is provided in deliverables D6.7 (in Annex) and D7.2 respectively. The list of ontologies used for this mapping is shown in Table 88.

Table 88: Ontologies used for BIM4Occupant tool

BIM4EEB Ontologies	Non-BIM4EEB ontologies
Digital Construction Entities (dice)	Semantic Sensor Networks (ssn) Sensor, Observation, Sample and Actuator (sosa)
Digital Construction Variables (dicv)	Smart Appliances REFerence (saref), SAREF ontology for Buildings (s4bldg)
Digital Construction Information (dici)	
Digital Construction Processes (dicp)	
Digital Construction Agents (dica)	
Digital Construction Occupancy (dicob)	
Digital Construction Energy (dices)	

The BIM4Occupant support the REST interface and can access the data from BIMMS. The Table 89 shows the system criteria supported by the BIM4Occupant tool.

Table 89: The addressed criteria in BIM4Occupants in relation to linked data

System Criteria	Implementation Details
Each system that provides data to other systems must:	
P1. Implement the Linked Data Principles (Berners-Lee, 2006)	
P2. Represent the shared data (maybe converted from native data) as RDF graphs with links to additional data	
P3. Support the specified interfaces	
P4. Use the specified set of shared ontologies to define the types/properties of entities in shared data	
Each system that consumes data from other systems must:	
C1. Be able to access the data using specified identifiers, query language, and interfaces	X
C2. Be able to parse the received data and query results	X
C3. Use the specified set of shared ontologies to interpret the types/properties of objects in shared data	X
Identifiers	HTTP URIs, using the HTTPS protocol
Data formats	
Interfaces	REST API (part of BIMMS)
Use of Existing ontologies	X

7.5 BIM4EEB BIMcpd tool

The BIMcpd (c - Constraint Checking; p - Performance evaluation; d - Data Management) toolset contains several distinct intuitive applications. These tools' components were designed to ensure minimum complications for users while maximizing the outputs of each tool. (Daniotti, et al., 2020).

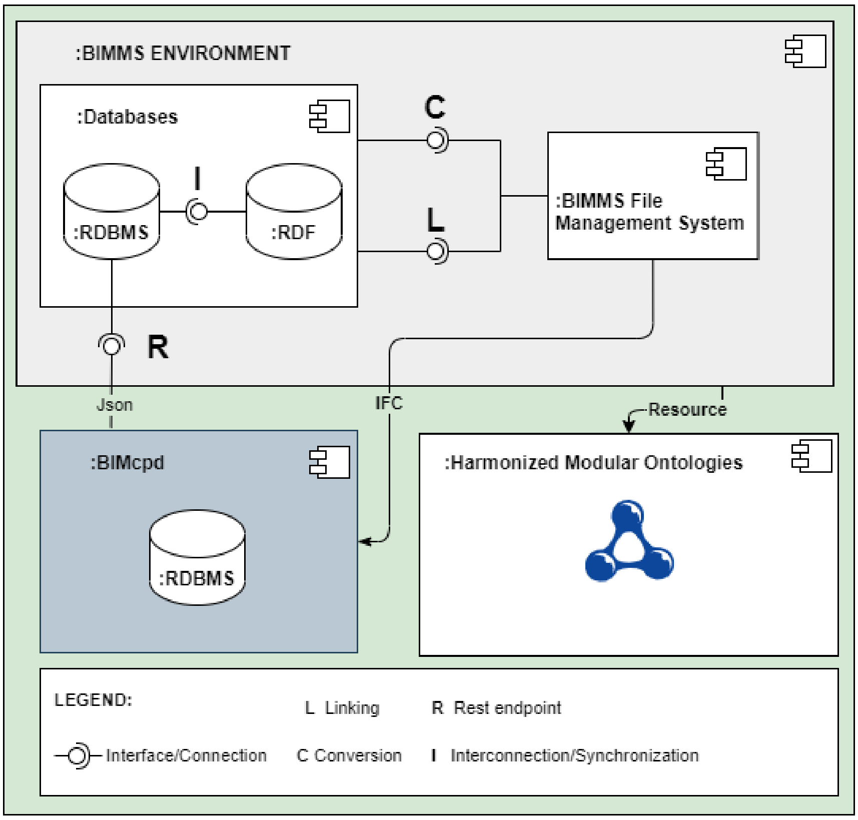

Figure 27: Usage of ontologies for BIMcpd

The BIMcpd will analyse an imported IFC file from the BIMMS, will map a number of elements related with the following ontologies within this file, and display it on-screen. The ontologies considered for the above mentioned IFC elements (if existing within that IFC file) are shown in the Table 90.

Table 90: Ontologies used for BIMcpd tool

BIM4EEB Ontologies	Non-BIM4EEB ontologies
Digital Construction Entities (dice)	
Digital Construction Occupancy (dicob)	
Digital Construction Lifecycle (dicl)	

The knowledge of this ontological preciseness will respectively allow users to:

- prioritise those BIMcpd' recommended positions for HVAC, lighting and other devices' (mentioned in the D6.3-5 report) for installation during the renovation processes. Moreover, for diagnosis purposes it is of interest to understand what number of building services are available in a dedicated space. Furthermore, knowledge about the "grouping" of building services components in "supply circuits" or "sub-circuits" is of information interest.
- better analyze data from sensors, meters and other additional sources (e.g. energy bills, weather data etc.); this is when aiming to reliably evaluate the comfort and other efficiency parameters of

integrated building systems and/or their components before and after renovation. Additionally, it is not only important to access information about those building elements “enclosing” the spatial structures (e.g. walls, slabs, etc.), but also to get a better knowledge about those elements “interfacing” with the “external world”, i.e. sensors and meters. Last but not least, descriptive data of building services components are relevant for performance evaluation.

- manage data gathered from the abovementioned sources and to create new meaningful data sets for sharing with other tools, for example, during the renovation process and highlighting the necessity in deeper BIM precision and accuracy (e.g. LoD). Thus, often so called dynamic data needs to be acquired and recomposed. This is the data describing the status of a “Spatial Structural Element” or a “building services component” (e.g. temperature, humidity, electricity consumption, status “on/off”). The origin of this data can be either a “monitoring device” (e.g. sensor, meter, actuator) or a “virtual devices”, but resulting dataset could be used, for example, by an additional simulation system.

Finally, it is necessary to mention that BIMcpd will be continually assessed and improved during the whole BIM4EEB project and even further on, so any ontological and other relative improvements and corrections that arise will be noted, rectified, and updated to include any additional elements not previously covered for all three BIMcpd interconnected components.

The criteria defined in the frame work are considered in the BIMcpd and able to support the these criteria’s. Those details are mentioned in the Table 91.

Table 91: The addressed criteria in BIMcpd in relation to linked data

System Criteria	**Implementation details**
Each system that provides data to other systems must:	
P1. Implement the Linked Data Principles (Berners-Lee, 2006)	
P2. Represent the shared data (maybe converted from native data) as RDF graphs with links to additional data	
P3. Support the specified interfaces	
P4. Use the specified set of shared ontologies to define the types/properties of entities in shared data	
Each system that consumes data from other systems must:	
C1. Be able to access the data using specified identifiers, query language, and interfaces	X
C2. Be able to parse the received data and query results	X
C3. Use the specified set of shared ontologies to interpret the types/properties of objects in shared data	X
Identifiers	HTTP URIs, using the HTTPS protocol
Data formats	
Interfaces	SPARQL Endpoint, REST API (part of BIMMS)
Use of Existing ontologies	

8 Conclusions

The main value of this deliverable is BIM4EEB linked data modelling and sharing framework. As part of the framework, practical workability of BIM4EEB ontologies were analysed by considering the different criterias mentioned for the ontologies in the BIM4EEB D3.1 deliverable. Later, ontology consistency is addressed by developing the ontology alignments and carrying the evaluation process for ontologies. Finally, exploitation of ontologies in BIM4EEB tools were addressed.The final outcome of this deliverable is harmonized modular ontology suite, which is DiCon ontology suite[2].

8.1.1 Potential Usage of Ontology Modules in BIM4EEB

In this last section, we provide an overview outlining the potential of the developed ontologies for future use in the subsequent BIM4EEB work packages. The developed ontologies mainly used in the BIM4EEB Work Package (WP) 4, 6, and 7. The tools developed in this work packages and which use linked data framework are BIMMS, BIMplanner, BIMeaser, BIM4Occupants, and BIMcpd tool. The Table 92 list the BIM4EEB tools and the ontologies used by this tools. It has to be noted that BIMMS has a capability to take any type of ontology as a resource. However here we listed only the ontologies which are used in the BIMMS for linked data graphs to store rdf data.

Table 92: Usage of ontologies in BIM4EEB tools

Tools / BIM4EEB Ontologies/ Vocabularies	BIMMS	BIMPlanner	BIMeaser	BIM4Occupants	BIMcpd
Contexts		X	X		
Variables	X	X	X	X	
Entities	X	X	X	X	X
Processes	X	X		X	
Agents		X		X	
Information	X	X	X	X	
Materials		X	X		
Occupancy	X	X		X	X
Lifecycle		X			X
Energy		X	X		
Units		X	X		
Levels		X			
Stages		X			
Non BIM4EEB	X	X	X	X	

8.1.2 Cooperation and Demarcation to other Projects

To reuse existing ontology work and to avoid creating ontologies from scratch, it was decided in BIM4EEB to build its ontology work onto the basis of the ontologies developed in the DiCtion project. DiCtion ("Digitalizing Construction Workflows") was a national Finnish research project from 2018-2020 funded by Business Finland. The project had strong industrial participation from the most advanced construction and renovation companies in Finland (Fira, Sweco, and Bonava – advanced companies even on the worldwide scale), large pre-fabricators (Ruukki and Parma Consolis), a main player in the BIM software industry (Trimble), and world-class academic research on construction management (Aalto University and VTT, the State Research Centre of Finland). The research was led by professor Olli Seppänen from Aalto University, the father of the Vico Schedule Planner construction management system.

DiCtion developed a set of ontology modules to support modern BIM/IoT-based construction management methods and practices, including Location-Based Management System, Takt-time planning, and the Last Planner™. DiCtion itself was based on earlier projects DRUMBEAT ("Distributed Reactive Management for Web-based Construction Lifecycle") on Linked Building Data, and ReCap ("Reality capture") and iCons ("Intelligent Construction Site") on the use of sensor data in construction. DiCtion held a number of workshops on construction management ontologies and finally specified four ontology modules: Objects, Organizations, Planning, and Monitoring (Figure 28).

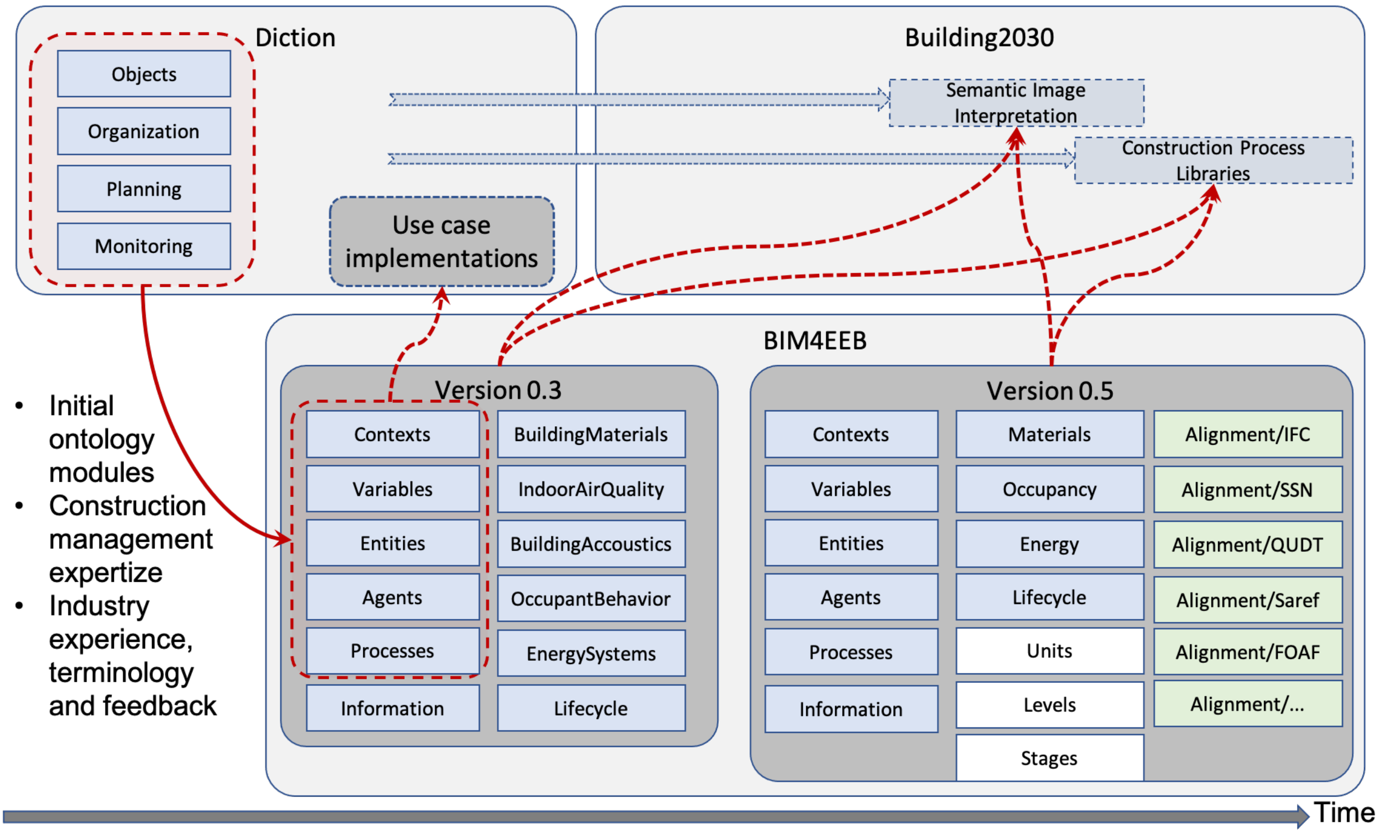

Figure 28: Relations of ontology work of BIM4EEB to other projects

When BIM4EEB began and the state-of-the-art survey was made – as reported in Deliverable 3.1 – it became apparent that Diction presents the most advanced and ambitious ontology work in the construction management domain, and that the ontologies were well in line with the goals of BIM4EEB. Since the ontology development in DiCtion was already winding down, an agreement with DiCtion was made that

BIM4EEB will continue the active development of those ontologies, instead of creating similar ontologies from the scratch. In the transition point in Spring 2019, the ontologies were named Digital Construction Ontologies, initially with the acronym DICO and later (due to a name conflict with Defense Intelligence Core Ontologies) with DiCon, and a GitHub organization for ‘digitalconstruction’ was created. Since both projects wanted to publish the ontologies with a permissive Creative Commons Attribution Licence, there was no concern that any conflicts would ensue.

During the transition, a new set of ontology modules were envisioned – Contexts, Variables, Entities, Agents, Processes, and Information – to extend and enhance the previous modules, as shown in Figure 28. BIM4EEB took the responsibility for the ontology definition work and the expansion of the ontology suite with additional ontologies, specifically focused on energy renovations: BuildingMaterials, IndoorAirQuality, BuildingAcoustics, OccupantBehavior, and Lifecycle. The definition work on all these modules was done completely within BIM4EEB and resulted in version 0.3 of Digital Construction Ontologies. Subsequently, BIM4EEB published a new, extended, refactored, and reorganized version 0.5, based on the vertical and horizontal segmentation approach of the Semantic Sensor Network Ontology. In version 0.5 the modules are the following: Contexts, Variables, Entities, Processes, Agents, Information, Materials, Occupancy, Lifecycle, Energy.

9 Bibliography

[ISO 16739, 2013] ISO 2013: ISO 16739: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries, Geneva: International Organization for Standardization.

[Bonduel, 2018] Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen. M., Klein, Ralf., 2018: The IFC to linked building data converter-current status. 6th Linked Data in Architecture and Construction Workshop, Volume 2159, pp. 34--43.

[Beetz, 2009] Beetz, J., Jos, V. L., Bauke, D. V., 2009: IfcOWL: A case of transforming EXPRESS schemas into ontologies. Ai Edam, Volume 23, pp. 89--101

[Smith, 2008] Smith, B., Brochhausen, M., 2008: Establishing and harmonizing ontologies in an interdisciplinary health care and clinical research environment.

[Euzenat, 2011] Euzenat, J., Meilicke, Ch., Stuckenschmidt, H., Shvaiko, P., Trojahn C., 2011. Ontology alignment evaluation initiative: six years of experience. Journal on data semantics XV, pp. 158--192.

[Said Rabah, 2013] Said Rabah, A., Shikun, Z.,2013: Comparative Analysisof Ontology Alignment Methodology.

[Otero-Cerdeira, 2015] Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A., 2015: Ontology matching: A literature review. Expert Systems with Applications, 42(2), pp.949-971.

[Vrandecic, 2009] Vrandecic, D., 2009: Ontology evaluation. In: Handbook on Ontologies | International Handbooks in Information Systems. Berlin, Heidelberg: Springer, p. 293 to 313.

[Balzert, 2000] Balzert, H., 2000: Lehrbuch der Software-Technik; vol. 1 and 2: Spektrum Akademischer Verlag; Heidelberg, Berlin.

[Bruegge, 2013] Bruegge, B., Dutoit A. H., 2013: Object-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International Edition, 3. Auflage, Prentice Hall, Boston, London, Munich, Tokyo, etc.

[Stuckenschmidt, 2009] Stuckenschmidt, H., Parent, Ch., Spaccapietra, S., 2009: “Modular Ontologiesoncepts, Theories and Techniques for Knowledge Modularization. Springer Lecture Notes in Computer Science / Theoretical Computer Science vol. 5445: Berlim, Heidelberg.

[Filipowska, 2009] Filipowska A., Hepp M., Kaczmarek M., Markovic I., 2009: Organisational Ontology Framework for Semantic Business Process Management. In: Abramowicz W. (eds) Business Information Systems. BIS 2009. Lecture Notes in Business Information Processing, vol 21. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01190-0_1.

[Hlomani, 2014] Hlomani, H., Stacey, D., 2014: Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. Semantic Web Journal, 1(5), pp.1-11..

[Raad, 2015] Raad, J. and Cruz, C., 2015: A survey on ontology evaluation methods. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.

[Brank, 2005] Brank, J., Grobelnik, M., Mladenic, D., 2005: A survey of ontology evaluation techniques. In Proceedings of the conference on data mining and data warehouses (SiKDD 2005) (Vol. 17). Citeseer Ljubljana, Slovenia.

[Ensan, 2013] Ensan, F., Du, W., 2013: A semantic metrics suite for evaluating modular ontologies. Information Systems, 38(5), pp.745-770.

[Alessandro, 2020] Alessandro, V., Davide, M., Jacopo, C., Diego, F. 2020: The BIM Management System: A Common Data Environment Using Linked Data to Support the Efficient Renovation in Buildings. In Proceedings 65 (1), p. 18. DOI: 10.3390/proceedings2020065018.

[Rasmussen, 2019] Rasmussen, M.H., Lefrançois, M., Pauwels, P., Hviid, C.A., Karlshøj, J., 2019: Managing interrelated project information in AEC Knowledge Graphs. Automation in Construction, 108, p.102956.

[ISO 21597-1] ISO 21597-1:2020: Information container for linked document delivery — Exchange specification — Part 1: Container

[Compton, 2012] Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A. Huang, V., 2012: The SSN ontology of the W3C semantic sensor network incubator group. Journal of Web Semantics, 17, pp.25-32.

[Bernardo, 2007] Bernardo G., Ian H., Yevgeny K., Uli S.: A logical framework for modular integration of ontologies. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). 2007.

[Mathieu, 2009] Mathieu d'A., Anne S., Heiner S., Marta S., 2009: Criteria and Evaluation for Ontology Modularization Techniques. In Heiner Stuckenschmidt, Christine Parent, Stefano Spaccapietra (Eds.): Modular Ontologies, vol. 5445. Berlin, Heidelberg: Springer Nature (Lecture Notes in Computer Science), pp. 67–89.

[Sarra, 2012] Sarra B.A., Andreas S., Thomas M., Mathieu d'A., 2012: Characterizing Modular Ontologies. In CEUR Workshop Proceedings 875.

[Marc, 2007] Marc E., 2007: Ontology Alignment. Bridging the Semantic Gap. Boston, MA: Springer US (Semantic Web and Beyond, Computing for Human Experience, 4).

[Inès, 2021] Inès O., Sadok B.Y., Gayo D., 2021: Ontology Integration: Approaches and Challenging Issues. In Information Fusion 71, pp. 38–63. DOI: 10.1016/j.inffus.2021.01.007.

[ISO 19650-1 2018] ISO 19650-1:2018: Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles

[Daniotti, 2020] Daniotti, B. et al., 2020. Workshop: BIM4EEB: A BIM-Based Toolkit for Efficient rEnovation in Buildings.

[Shemeikka, 2020] Shemeikka, J., Vesanen, T., Hasan, A., Mätäsniemi, T., 2020: Early Stage Energy Refurbishment Assessment Tool for Buildings Using High-End BIM Data: Benefits and Challenges. MDPI.

[Törmä, 2020] Törmä, S. et al., 2020. Towards BIM-Enhanced Renovation Management Tools with Support to Stakeholder Interaction.

[Uschold, 1996] Uschold, M. and Gruninger, M., 1996. Ontologies: Principles, methods and applications. The knowledge engineering review, 11(2), pp. 93--136.

[Wisniewski, 2018] Wisniewski, D., Potoniec, J., Lawrynowicz, A., Keet, C.M., 2018: Competency questions and SPARQL-OWL queries dataset and analysis. arXiv preprint arXiv:1811.09529.

[Ren, 2014] Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., Van Deemter, K., Stevens, R., 2014: Towards competency question-driven ontology authoring. In European Semantic Web Conference (pp. 752-767). Springer, Cham.

[Georg F., 2017] Schneider, G F., 2017: "Ontology, Towards Aligning Domain Ontologies with the Building Topology."

[org, 2014] The Organization Ontology: https://www.w3.org/TR/vocab-org/.

10 ANNEX I: Documentation of Alignments between DiCon ontologies and external ontologies

10.1.1 Alignment between the context and external ontologies

Table 93: The alignment of dicc ontology with external ontologies

Subject	Predicate	Object
ifc:IfcContext	rdfs:subClassOf	dicc:Context

10.1.2 Alignment between the variables ontology with external ontology

Table 94: The alignment of dicv ontology with external ontologies

Subject	Predicate	Object
dicv:QuantitativeProperty	owl:equivalentClass	qudt:Quantity
dicv:QuantityKind	owl:equivalentClass	qudt:QuantityKind
dicv:QuantitativeState	owl:equivalentClass	qudt:QuantityValue
dicv:Unit	owl:equivalentClass	qudt:Unit
dicv:Subject	owl:subClassOf	qudt:Quantifiable
dicv:PropertyState	owl:equivalentClass	opm:PropertyState
dicv:Subject	owl:equivalentClass	sosa:FeatureOfInterest
dicv:Property	owl:equivalentClass	ssn:Property
sosa:ObservableProperty	rdfs:subClassOf	dicv:Property
dicv:QuantitativeState	owl:equivalentClass	sosa:Result
dicv:Subject	owl:equivalentClass	saref:FeatureOfInterest
dicv:QuantitativeProperty	owl:equivalentClass	saref:Property
dicv:QuantitativeState	owl:equivalentClass	saref:Measurement
saref:UnitOfMeasure	rdfs:subClassOf	dicv:Unit
dicv:hasPropertyState	rdfs:inverseOf	saref:relatesToProperty
dicv:hasQuantityKind	owl:equivalentProperty	qudt:hasQuantityKind
qudt:quantityValue	rdfs:subPropertyOf	dicv:hasPropertyState
dicv:hasUnit	owl:equivalentProperty	qudt:unit
dicv:hasValue	owl:equivalentProperty	qudt:value
qudt:hasQuantity	owl:subPropertyOf	dicv:hasProperty
dicv:Property	owl:equivalentClass	opm:Property, ifc:IfcProperty
dicv:hasPropertyState	owl:equivalentProperty	opm:hasPropertyState
saref:hasProperty	rdfs:subPropertyOf	dicv:hasProperty
saref:isMeasuredIn	rdfs:subProperty	dicv:hasUnit
dicv:hasValue	owl:equivalentProperty	saref:hasValue

10.1.3 Alignment between entities ontology with other ontologies

Table 95: The alignment of dice ontology with external ontologies

Subject	Predicate	Object
ifc:IfcSite	rdfs:subClassOf	dice:Location
dice:RealEstate	rdfs:subClassOf	ifc:IfcSite
ifc:IfcBuilding	rdfs:subClassOf	dice:Building
ifc:IfcElement	rdfs:subClassOf	dice:BuildingObject
ifc:SpatialElement	rdfs:subClassOf	dice:Location
ifc:IfcZone	rdfs:subClassOf	dice:Location
ifc:IfcAsset	rdfs:subClassOf	dice:Asset
ifc:IfcSensor	rdfs:subClassOf	dice:Sensor
ifc:IfcConstructionEquipmentResource	rdfs:subClassOf	dice:Equipment
ifc:IfcTask	rdfs:subClassOf	dice:Activity
dice:Group	owl:equivalentClass	ifc:IfcGroup
dice:Location	owl:equivalentClass	bot:Zone
dice:Building	owl:equivalentClass	bot:Building
dice:BuildingObject	owl:equivalentClass	bot:Element
dice:Sensor	owl:equivalentClass	sosa:Sensor
ssn:Platform	rdfs:subClassOf	dice:MaterialEntity
dice:Device	owl:equivalentClass	saref:Device
dice:Sensor	owl:equivalentClass	saref:Sensor
s4bldg:Sensor	rdfs:subClassOf	dice:Sensor
s4bldg:BuildingSpace	rdfs:subClassOf	dice:Location
dice:Function	owl:equivalentClass	saref:Function
core:Apartment	rdfs:subClassOf	dice:ResidentialUnit
dice:Building	owl:equivalentClass	core:Building
dice:BuildingObject	rdfs:subClassOf	core:BuildingComponent
dice:BuildingUnit	rdfs:subClassOf	core:BuildingComponent
dice:Group	owl:equivalentClass	core:Collection
dice:Device	owl:equivalentClass	core:Device
dice:RealEstate	owl:equivalentClass	core:RealEstate
dice:Location	rdfs:subClassOf	core:Room
dice:Sensor	owl:equivalentClass	core:Sensor
dice:TimeInterval	owl:equivalentClass	time:Interval
dice:TimeInstant	owl:equivalentClass	time:Instant
dice:TimeDuration	owl:equivalentClass	time:Duration
org:Role	rdfs:subClassOf	dice:Role
dicv:Property	rdfs:subClassOf	prov:Entity
dice:hasSubLocation	owl:equivalentProperty	bot:containsZone
dice:isLocationOf	owl:equivalentProperty	bot:hasELement
bot:hasSubElement	rdfs:subPropertyOf	dice:hasContinuantPartAtAllTimes
dice:hasStart	owl:equivalentProperty	time:hasBeginning
dice:hasEnd	owl:equivalentProperty	time:hasEnd
foaf:member	rdfs:subPropertyOf	dice:hasMember
dice:hasLongitude	rdfs:subPropertyOf	geo:long
dice:hasLatitude	rdfs:subPropertyOf	geo:lat
dice:hasAltitude	rdfs:subPropertyOf	geo:alt
dice:isLocatedIn	rdfs:subPropertyOf	geo:location

Table 96: The alignment of dice ontology with BFO ontology

Subject	Predicate	Object
dice:Entity	owl:equivalentClass	obo:BFO_0000001
dice:Continuant	owl:equivalentClass	obo:BFO_0000002
dice:Occurrent	owl:equivalentClass	obo:BFO_0000003
dice:IndependentContinuant	owl:equivalentClass	obo:BFO_0000004
dice:SpatialRegion	owl:equivalentClass	obo:BFO_0000006
dice:TemporalRegion	owl:equivalentClass	obo:BFO_0000008
dice:TwoDimensionalSpatialRegion	owl:equivalentClass	obo:BFO_0000009
dice:SpatiotemporalRegion	owl:equivalentClass	obo:BFO_0000011
dice:Process	owl:equivalentClass	obo:BFO_0000015
dice:Disposition	owl:equivalentClass	obo:BFO_0000016
dice:RealizableEntity	owl:equivalentClass	obo:BFO_0000017
dice:ZeroDimensionalSpatialRegion	owl:equivalentClass	obo:BFO_0000018
dice:Quality	owl:equivalentClass	obo:BFO_0000019
dice:SpecificallyDependentContinuant	owl:equivalentClass	obo:BFO_0000020
dice:Role	owl:equivalentClass	obo:BFO_0000023
dice:FiatObjectPart	owl:equivalentClass	obo:BFO_0000024
dice:OneDimensionalSpatialRegion	owl:equivalentClass	obo:BFO_0000026
dice:ObjectAggregate	owl:equivalentClass	obo:BFO_0000027
dice:ThreeDimensionalSpatialRegion	owl:equivalentClass	obo:BFO_0000028
dice:Site	owl:equivalentClass	obo:BFO_0000029
dice:Object	owl:equivalentClass	obo:BFO_0000030
dice:GenericallyDependentContinuant	owl:equivalentClass	obo:BFO_0000031
dice:Function	owl:equivalentClass	obo:BFO_0000034
dice:ProcessBoundary	owl:equivalentClass	obo:BFO_0000035
dice:OneDimensionalTemporalRegion	owl:equivalentClass	obo:BFO_0000038
dice:MaterialEntity	owl:equivalentClass	obo:BFO_0000040
dice:ContinuantFiatBoundary	owl:equivalentClass	obo:BFO_0000140
dice:ImmaterialEntity	owl:equivalentClass	obo:BFO_0000141
dice:FiatLine	owl:equivalentClass	obo:BFO_0000142
dice:RelationalQuality	owl:equivalentClass	obo:BFO_0000145
dice:FiatSurface	owl:equivalentClass	obo:BFO_0000146
dice:FiatPoint	owl:equivalentClass	obo:BFO_0000147
dice:ZeroDimensionalTemporalRegion	owl:equivalentClass	obo:BFO_0000148
dice:History	owl:equivalentClass	obo:BFO_0000182
dice:TimeInterval	owl:equivalentClass	obo:BFO_0000202
dice:TimeInstant	owl:equivalentClass	obo:BFO_0000203
dice:hasRealization	owl:equivalentProperty	obo:BFO_0000054
dice:realizes	owl:equivalentProperty	obo:BFO_0000055

Subject	Predicate	Object
dice:participatesIn	owl:equivalentProperty	obo:BFO_0000056
dice:hasParticipant	owl:equivalentProperty	obo:BFO_0000057
dice:isConcretizedBy	owl:equivalentProperty	obo:BFO_0000058
dice:concretizes	owl:equivalentProperty	obo:BFO_0000059
dice:precededBy	owl:equivalentProperty	obo:BFO_0000062
dice:precedes	owl:equivalentProperty	obo:BFO_0000063
dice:occursIn	owl:equivalentProperty	obo:BFO_0000066
dice:locatedInAtAllTimes	owl:equivalentProperty	obo:BFO_0000082
dice:genericallyDependsOn	owl:equivalentProperty	obo:BFO_0000084
dice:isCarrierOf	owl:equivalentProperty	obo:BFO_0000101
dice:existsAt	owl:equivalentProperty	obo:BFO_0000108
dice:hasContinuantPartAtAllTimes	owl:equivalentProperty	obo:BFO_0000110
dice:hasProperContinuantPartAtAllTimes	owl:equivalentProperty	obo:BFO_0000111
dice:hasMaterialBasisAtAllTimes	owl:equivalentProperty	obo:BFO_0000113
dice:hasMemberPart	owl:equivalentProperty	obo:BFO_0000115
dice:hasOccurrentPart	owl:equivalentProperty	obo:BFO_0000117
dice:hasProperOccurrentPart	owl:equivalentProperty	obo:BFO_0000118
dice:hasTemporalPart	owl:equivalentProperty	obo:BFO_0000121
dice:locationOf	owl:equivalentProperty	obo:BFO_0000124
dice:materialBasisOf	owl:equivalentProperty	obo:BFO_0000127
dice:memberPartOf	owl:equivalentProperty	obo:BFO_0000129
dice:occurrentPartOf	owl:equivalentProperty	obo:BFO_0000132
dice:properTemporalPartOf	owl:equivalentProperty	obo:BFO_0000136
dice:properContinuantPartOfAtAllTimes	owl:equivalentProperty	obo:BFO_0000137
dice:properOccurrentPartOf	owl:equivalentProperty	obo:BFO_0000138
dice:temporalPartOf	owl:equivalentProperty	obo:BFO_0000139
dice:temporallyProjectsOnto	owl:equivalentProperty	obo:BFO_0000153
dice:materialBasisOfAtAllTimes	owl:equivalentProperty	obo:BFO_0000163
dice:concretizesAtAllTimes	owl:equivalentProperty	obo:BFO_0000164
dice:isConcretizedByAtAllTimes	owl:equivalentProperty	obo:BFO_0000165
dice:participatesInAtAllTimes	owl:equivalentProperty	obo:BFO_0000166
dice:hasParticipantAtAllTimes	owl:equivalentProperty	obo:BFO_0000167
dice:locationOfAtAllTimes	owl:equivalentProperty	obo:BFO_0000170
dice:locatedIn	owl:equivalentProperty	obo:BFO_0000171
dice:hasMemberPartAtAllTimes	owl:equivalentProperty	obo:BFO_0000172
dice:memberPartOfAtAllTimes	owl:equivalentProperty	obo:BFO_0000173
dice:hasProperContinuantPart	owl:equivalentProperty	obo:BFO_0000174
dice:properContinuantPartOf	owl:equivalentProperty	obo:BFO_0000175
dice:continuantPartOf	owl:equivalentProperty	obo:BFO_0000176
dice:continuantPartOfAtAllTimes	owl:equivalentProperty	obo:BFO_0000177

Subject	Predicate	Object
dice:hasContinuantPart	owl:equivalentProperty	obo:BFO_0000178
dice:hasProperTemporalPart	owl:equivalentProperty	obo:BFO_0000181
dice:environs	owl:equivalentProperty	obo:BFO_0000183
dice:historyOf	owl:equivalentProperty	obo:BFO_0000184
dice:hasHistory	owl:equivalentProperty	obo:BFO_0000185
dice:specificallyDependedOnBy	owl:equivalentProperty	obo:BFO_0000194
dice:specificallyDependsOn	owl:equivalentProperty	obo:BFO_0000195
dice:bearerOf	owl:equivalentProperty	obo:BFO_0000196
dice:inheresIn	owl:equivalentProperty	obo:BFO_0000197
dice:occupiesTemporalRegion	owl:equivalentProperty	obo:BFO_0000199
dice:occupiesSpatiotemporalRegion	owl:equivalentProperty	obo:BFO_0000200
dice:occupiesSpatialRegion	owl:equivalentProperty	obo:BFO_0000210
dice:occupiesSpatialRegionAtAllTimes	owl:equivalentProperty	obo:BFO_0000211
dice:spatiallyProjectsOnto	owl:equivalentProperty	obo:BFO_0000216
dice:spatiallyProjectsOntoAtAllTimes	owl:equivalentProperty	obo:BFO_0000217
dice:hasMaterialBasis	owl:equivalentProperty	obo:BFO_0000218
dice:genericallyDependsOnAtAllTimes	owl:equivalentProperty	obo:BFO_0000219
dice:isCarrierOfAtAllTimes	owl:equivalentProperty	obo:BFO_0000220
dice:firstInstantOf	owl:equivalentProperty	obo:BFO_0000221
dice:hasFirstInstant	owl:equivalentProperty	obo:BFO_0000222
dice:lastInstantOf	owl:equivalentProperty	obo:BFO_0000223
dice:hasLastInstant	owl:equivalentProperty	obo:BFO_0000224

10.1.4 Alignment between processes ontology with other ontologies

Table 97: The alignment of dicp ontology with external ontologies

Subject	Predicate	Object
dicp:Resource	owl:equivalentClass	ifc:IfcResource
ifc:IfcConstructionResource	rdfs:subClassOf	dicp:Resource
ifc:IfcProject	rdfs:subClassOf	dicp:Project
dicp:Observation	owl:equivalentClass	sosa:Observation
dicp:Actuation	owl:equivalentClass	sosa:Actuation
dicp:Service	owl:equivalentClass	saref:Service
dicp:Actuation	owl:equivalentClass	device:Actuation
dicp:Observation	owl:equivalentClass	core:Observation
dicp:Service	owl:equivalentClass	core:Service
dicp:Project	rdfs:subClassOf	foaf:Project
dicp:Activity	rdfs:subClassOf	prov:Activity
sosa:observedProperty	rdfs:subPropertyOf	dicp:hasObservedProperty
sosa:actsOnProperty	rdfs:subPropertyOf	dicp:actsOnProperty
dicp:hasObservedResult	owl:equivalentProperty	sosa:hasResult

10.1.5 Alignment between the agents and external ontologies

Table 98: The alignment of dica ontology with external ontologies

Subject	Predicate	Object
dica:Actor	owl:equivalentClass	ifc:IfcActor
dica:Person	owl:equivalentClass	ifc:IfcPerson
dica:Organization	owl:equivalentClass	ifc:IfcOrganization
dica:Occupant	owl:equivalentClass	ifc:IfcOccupant
ifc:IfcCrewResource	rdfs:subClassOf	dica:TaskTeam
dica:Agent	owl:equivalentClass	ct:Party
dica:Agent	owl:equivalentClass	core:Agent
dica:Team	owl:equivalentClass	agents:Group
dica:Organization	owl:equivalentClass	agents:Organization
dica:Person	owl:equivalentClass	agents:Person
dica:Agent	owl:equivalentClass	foaf:Agent
dica:Person	owl:equivalentClass	foaf:Person
dica:Organization	owl:equivalentClass	foaf:Organization
dica:Team	owl:equivalentClass	foaf:Group
dica:Organization	owl:equivalentClass	org:Organization
dica:LegalEntity	owl:equivalentClass	org:FormalOrganization
org:hasMember	rdfs:subPropertyOf	dica:hasOrganizationPart

10.1.6 Alignment between the information and external ontologies

Table 99: The alignment of dici ontology with external ontologies

Subject	Predicate	Object
ifc:IfcApproval	rdfs:subClassOf	dici:InformationContentEntity
ifc:IfcPermit	rdfs:subClassOf	dici:InformationContentEntity
ifc:IfcObjective	rdfs:subClassOf	dici:Goal
ifc:IfcWorkPlan	rdfs:subClassOf	dici:Plan
ifc:IfcWorkSchedule	rdfs:subClassOf	dici:Plan
Ifcowl:IfcEvent	rdfs:subClassOf	dici:Event
dici:informationModel	rdfs:subClassOf	dcat:Catalog
dici:DataSet	rdfs:subClassOf	dcat:DataSet
ct:ContainerDescription	rdfs:subClassOf	dici:InformationModel
ct:Document	rdfs:subClassOf	dici:Dataset
ct:Linkset	rdfs:subClassOf	dici:CrossFileLinkset
dici:Image	owl:equivalentClass	foaf:Image
foaf:Document	rdfs:subClassOf	dici:InformationContentEntity
dici:InformationContentEntity	rdfs:subClassOf	prov:Entity
ct:containsDocument	rdfs:subPropertyOf	dici:includesContainer
ct:belongsToContainer	rdfs:subPropertyOf	dici:includedInModel
ct:containsLinkset	rdfs:subPropertyOf	dici:includesContainer
ct:containedInContainer	rdfs:subPropertyOf	dici:includedInModel
core:Information	rdfs:subClassOf	dici:InformationContentEntity
ct:containsDocument	rdfs:subPropertyOf	dici:includesContainer
ct:belongsToContainer	rdfs:subPropertyOf	dici:includedInModel
ct:containsLinkset	rdfs:subPropertyOf	dici:includesContainer
ct:containedInContainer	rdfs:subPropertyOf	dici:includedInModel
core:Information	rdfs:subClassOf	dici:InformationContentEntity

10.1.7 Alignment between the material ontology and external ontologies

Table 100: The alignment of dicm ontology with external ontologies

Subject	Predicate	Object
dicm:Material	owl:equivalentClass	ifc:IfcMaterial
dicm:MaterialObjectStructure	owl:equivalentClass	ifc:IfcMaterialDefinition
dicm:Layer	owl:equivalentClass	ifc:MaterialLayer
dicm:LayerSet	owl:equivalentClass	ifc:IfcMaterialLayerSet
dicm:hasLayer	owl:equivalentProperty	ifc:materialLayers_IfcMaterialLayerSet
dicm:hasLayerSet	owl:equivalentProperty	ifc:forLayerSet_IfcMaterialLayerSetUsage
dicm:hasMaterial	owl:equivalentProperty	ifc: material_IfcMaterialLayer
dicm:adjacentElement	owl:equivalentProperty	bot:adjacentElement

10.1.8 Alignment between the occupancy ontology and external ontologies

Table 101: The alignment of dicob ontology with external ontologies

Subject	Predicate	Object
dicob:AirQualitySensor	rdfs:subClassOf	ifc:IfcSensor, sosa:Sensor, saref:Sensor, core:Sensor
dicob:HumiditySensor	rdfs:subClassOf	ifc:IfcSensor, sosa:Sensor, saref:Sensor, core:Sensor
dicob:LuminanceSensor	rdfs:subClassOf	ifc:IfcSensor, sosa:Sensor, saref:Sensor, core:Sensor
dicob:NoiseSensor	rdfs:subClassOf	ifc:IfcSensor, sosa:Sensor, saref:Sensor, core:Sensor
dicob:TemperatureSensor	rdfs:subClassOf	ifc:IfcSensor, sosa:Sensor, saref:Sensor, core:Sensor
dicob:TemperatureSensor	owl:equivalentClass	saref:TemperatureSensor
dicob:Ceiling	rdfs:subClassOf	ifc:IfcCovering
dicob:Door	owl:equivalentClass	ifc:IfcDoor
dicob:Door	rdfs:subClassOf	bot:Element, core:BuildingComponent
dicob:FacadeElement	rdfs:subClassOf	ifc:IfcBuildingElement, core:BuildingComponent
dicob:Floor	owl:equivalentClass	ifc:IfcSlab
dicob:Floor	rdfs:subClassOf	bot:Element, core:BuildingComponent
dicob:FloorCovering	rdfs:subClassOf	ifc:IfcCovering
dicob:Wall	owl:equivalentClass	ifc:IfcWall
dicob:Wall	rdfs:subClassOf	bot:Element, core:BuildingComponent

10.1.9 Alignment between the lifecycle ontology and external ontologies

Table 102: The alignment of dicl ontology with external ontologies

Subject	Predicate	Object
dicl:InformationConsumer	rdfs:subClassOf	ifc:IfcActorRole, org:Role
dicl:InformationProcessor	rdfs:subClassOf	Ifc:IfcActorRole, org:Role
dicl:InformationProvider	rdfs:subClassOf	Ifc:IfcActorRole, org:Role
dicl:InformationFlowRole	rdfs:subClassOf	Ifc:IfcActorRole, org:Role

10.1.10 Alignment between the energy ontology and external ontologies

Table 103: The alignment of dices ontology with external ontologies

Subject	Predicate	Object
dices:Actuator	owl:equivalentClass	Ifc:IfcActuator,Saref:Actuator
ifc:IfcElectricGenerator	rdfs:subClassOf	dices:Generator

10.1.11 Alignment between the DiCon and DiCtion ontologies

Table 104: The alignment of DiCon and DiCtion ontologies

Subject	Predicate	Object
dictioncob:DomainObject	rdfs:subClassOf	dice:Entity
dictioncob:Classification	rdfs:subClassOf	dice:Category

dictioncob:IdentifierType	rdfs:subClassOf	dice:Category
dictioncob:ClassificationScheme	rdfs:subClassOf	dice:Category
dictioncob:identifierLabel	rdfs:subPropertyOf	dice:hasLabel
dictioncob:identifierType	rdfs:subPropertyOf	dice:hasScope
dictioncob:classificationCode	rdfs:subPropertyOf	dice:hasLabel
ddictioncob:classifictionType	rdfs:subPropertyOf	dice:hasScope
dice:Group	owl:equivalentClass	dictioncob:Group
dictioncob:PhysicalEntity	rdfs:subClassOf	dice:IndependentContinuant
dice:Location	owl:equivalentClass	dictioncob:Location
dici:InformationContentEntity	rdfs:subClassOf	dictioncob:InformationEntity
dici:isAbout	rdfs:subPropertyOf	dictioncob:informationEntityAbout
dici:Contract	owl:equivalentClass	dictioncor:Contract
dici:hasContrator	owl:equivalentProperty	dictioncor:promisor
dici:hasClient	owl:equivalentProperty	dictioncor:promise
dici:hasClientObligation	rdfs:subPropertyOf	dictioncor:hasObligation
dici:hasContractorObligation	rdfs:subPropertyOf	dictioncor:hasObligation
dica:LegalPerson	owl:equivalentClass	dictioncor:Actor
dica:LegallyCompetentNaturalPerson	owl:equivalentClass	dictioncor:Person
dica:LegalEntity	owl:equivalentClass	dictioncor:Organization
dica:Corporation	owl:equivalentClass	dictioncor:Company
dice:Sensor	owl:equivalentClass	dictioncdg:Sensor
dicp:Resource	rdfs:subClassOf	dictioncpl:Resource
dictioncpl:hasCapability	rdfs:subPropertyOf	dice:hasCapability
dictioncpl:Capacity	rdfs:subClassOf	dice:CapabilityToWork
dictioncpl:Capability	rdfs:subClassOf	dice:Capability
dictioncpl:CapabilityType	rdfs:subClassOf	dice:Category
dictioncpl:ResourceType	rdfs:subClassOf	dice:Category
dictioncpl:ActivityType	rdfs:subClassOf	dice:Category
dicp:Activity	owl:equivalentClass	dictioncpl:Activity
dicp:hasObject	rdfs:subPropertyOf	dictioncpl:object
dicp:hasLocation	rdfs:subPropertyOf	dictioncpl:object
dicp:hasInformation	rdfs:subPropertyOf	dictioncpl:object
dictioncpl:Condition	rdfs:subClassOf	dicv:Constraint
dictioncpl:precondition	rdfs:subPropertyOf	dicp:hasCondition
dictioncpl:executionCondition	rdfs:subPropertyOf	dicp:hasCondition
dice:needsCapability	rdfs:subPropertyOf	dictioncpl:requires
dice:Capability	rdfs:subClassOf	dictioncpl:CapabilityRequirement

11 ANNEX II: Documentation of Alignments between external ontologies

11.1.1 Alignment of bot with ifcowl, saref4bldg ontologies

The Building Topology Ontology was developed with the intention to simplify the modelling of Spatial Structural Elements as provided in ifcOWL. Therefore, the alignment seems to be a straight forward exercise. The alignment between BOT with ifcOWL, saref4bldg as described in the work of *[Georg F., 2017]* is presented below.

Table 105: The alignment of bot with ifcowl, saref4bldg ontologies

Subject	Predicate	Object
bot:Building	owl:equivalentClass	ifc:IfcBuilding
bot:Site	owl:equivalentClass	ifc:IfcSite
bot:Storey	owl:equivalentClass	ifc:IfcBuildingStorey
bot:Space	owl:equivalentClass	ifc:IfcSpace
bot:Element	owl:equivalentClass	ifc:IfcElement
bot:Building	owl:equivalentClass	saref4bldg:Building
bot:Space	owl:equivalentClass	saref4bldg:BuildingSpace
bot:Element	owl:equivalentClass	saref4bldg:PhysicalObject
bot:hasSpace	rdfs:subPropertyOf	saref4bldg:hasSpace
bot:ContainsElement	owl:equivalentProperty	saref4bldg:contains

11.1.2 Alignment of foaf and org ontologies

Overlaps exists between the Friend of a Friend and the Organization Ontology. Alignment relevant within the BIM4EEB context are presented in the following table. The alignment is considered from the work published on w3c Organization ontology page [org, 2014].

Table 106: The alignment of foaf and org ontologies

Subject	Predicate	Object
org:Organization	owl:equivalentClass	foaf:Organization
org:hasMember	owl:equivalentProperty	foaf:member

11.1.3 Alignment of rec with ifcowl, bot ontologies

In this section we present alignments between the Real Estate Core ontologyand the Building Topology Ontology. Alignments are provided for selected Spatial Structure Elements and Distribution Control Devices.

Table 107: The alignment of rec with ifcowl, bot ontologies

Subject	Predicate	Object
core:Land	rdfs:subClassOf	ifc:IfcSite
core:Building	rdfs:subClassOf	ifc:IfcBuilding
core:Floor	rdfs:subClassOf	ifc:IfcSlab
core:Slab	rdfs:subClassOf	ifc:IfcSlab
core:Wall	rdfs:subClassOf	ifc:IfcWall
core:Device	rdfs:subClassOf	ifc:IfcDistributionElement
core:Actuator	owl:equivalentClass	ifc:IfcActuator
core:Sensor	owl:equivalentClass	ifc:IfcSensor
core:Building	owl:equivalentClass	bot:Building
core:Land	rdfs:subClassOf	bot:Site
core:Device	rdfs:subClassOf	bot:Element

12 ANNEX III Sample data for ontologies

12.1.1 Sample data for Digital Construction Contexts (dicc) ontology

For the evaluation of Contexts ontology, first the following data is inserted using SPARQL UPDATE:

```
Load <https://digitalconstruction.github.io/Contexts/v/0.5/contexts.ttl>
prefix dice: <https://w3id.org/digitalconstruction/0.5/Entities#>
prefix dicv: <https://w3id.org/digitalconstruction/0.5/Variables#>
prefix dicp: <https://w3id.org/digitalconstruction/0.5/processes#>
prefix dicc: <https://w3id.org/digitalconstruction/0.5/contexts#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/XML/1998/namespace>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix time: <http://www.w3.org/2006/time#>
prefix :     <http://example.com/id/>
INSERT DATA
{
 :R1 a dicc:Context ; dicc:hasContent :R1Graph .
 GRAPH :R1Graph
 {
   :Renovation1 a dicp:RenovationProject ;
              dicp:hasObject :Building .
}
 :B1 a dicc:Context ; dicc:hasContent :B1Graph .
 GRAPH :B1Graph
 {
   :Building1 a dice:Building ;
      dice:hasBuildingUnit :Apartment101, :Apartment102 .
   :Apartment101 a dice:ResidentialUnit .
   :Apartment102 a dice:ResidentialUnit ;
      dice:hasAdjacentElement :Apartment101 .
}
 :R1MP1 a dicc:Context ; dicc:hasContent :R1MP1Graph .
 GRAPH :R1MP1Graph
{
   :Renovation1 dicp:hasSubActivity :R1Design, :R1Construction, :R1Handover .
   :R1Design a dicp:Activity ; time:overlaps :R1Construction .
   :R1Construction a dicp:Activity ; time:before :R1Handover .
   :R1Handover a dicp:Activity
}
}
```

The dataset defines three contexts, :R1 for project data, :B1 for built asset data, and :R1MP1 for the master plan of the project. Each of the contexts has a associated named graph that contains their content.

dicc-cq1 How to store and manage datasets separately? (for versions or alternatives)

dicc-cq2 Is the given statement true in the given context?

Example Data:

A new independent context for a second version of the masterplan (:R1MP2) is created with SPARQL UPDATE as follows:

```
INSERT DATA
{
 :R1MP2 a dicc:Context ; dicc:hasContent :R1MP2Graph .
```

```
  GRAPH :R1MP2Graph
{
   :Renovation1    dicp:hasSubActivity    :R1Design,    :R1Procurement,
                 :R1Construction,  :R1Handover .
   :R1Design a dicp:Activity ; time:overlaps :R1Procurement .
   :R1Procurement a dicp:Activity ; time:overlaps :R1Construction .
   :R1Construction a dicp:Activity ; time:before :R1Handover .
   :R1Construction a dicp:Activity ;
               dicp:hasSubActivity :R1ReplaceWindows , :R1InstallHeatPump .
   :R1ReplaceWindows a dicp:Activity ;
                time:overlaps :R1InstallHeatPump  .
   :R1InstallHeatPump a dicp:Activity .
   :R1Handover a dicp:Activity
 }
}
```

Example Data:

```
WITH :R1MP2Graph
DELETE { ?x time:overlaps ?y }
INSERT { ?x time:before   ?y }
WHERE  { ?x time:overlaps ?y  }
```

Finally the context :R1MP2 can be completely removed by the following SPARQL UPDATE forms:

```
PREFIX : <http://example.com/id/>
DELETE { :R1MP2 ?p ?o }
WHERE { :R1MP2 ?p ?o }
PREFIX : <http://example.com/id/>
DROP GRAPH :R1MP2Graph
```

dicc-cq3 What statements hold (are true) in the given context?

Example Data:

To define what information is relevant in the current state, we define the :DefaultContextSet that contains the active contexts :R1, :B1 and :R1MP2:

```
INSERT DATA {  :DefaultContextSet dicc:hasActiveContext :R1 , :B1 , :R1MP2 .  }
```

12.1.2 Sample data for Digital Construction Variables (dicv) ontology

Initialize by loading the Variables ontology:

```
LOAD <https://w3id.org/digitalcostruction/0.5/Variables/variables.ttl>
prefix dice: <https://w3id.org/digitalconstruction/0.5/Entities#>
prefix dicv: <https://w3id.org/digitalconstruction/0.5/Variables#>
prefix dicp: <https://w3id.org/digitalconstruction/0.5/processes#>
prefix dicc: <https://w3id.org/digitalconstruction/0.5/contexts#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/XML/1998/namespace>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix unit: <http://qudt.org/2.1/vocab/unit/>
prefix quantitykind: <http://qudt.org/vocab/quantitykind/>
prefix :     <http://example.com/id/>
```

dicv-cq1 What are all the values of a property of an entity over time? (evolution of the value)

dicv-cq2 What is the quantity kind and unit of a quantitative property?

Example data:

```
INSERT DATA
{
 :A1 a dicp:Activity ;
   dicv:hasProperty [ a dicv:QuantitativeProperty ;
                      dicv:isPropertyFor dicp:hasActivityCost ;
                      dicv:hasPropertyState
                        [ a dicc:QuantitativeState ;
                             dicv:hasValue "800"^^xsd:decimal ;
                             dicv:hasUnit unit:Euro ;
                             dicv:hasQuantityKind quantitykind:currency ;
                             dicv:hasTimeOfCreation "2021-05-15T12:00:00"^^xsd:dateTime ] ,
                        [ a dicv:QuantitativeState ;
                             dicv:hasValue "840"^^xsd:decimal ;
                             dicv:hasUnit unit:Euro ;
                             dicv:hasQuantityKind quantitykind:currency ;
                          dicv:hasTimeOfCreation "2021-05-20T12:00:00"^^xsd:dateTime]]
                      }
Add value:
INSERT
{
 ?p dicv:hasPropertyState
    [ a dicv:QuantitativeState ;
         dicv:hasValue "860"^^xsd:decimal ;
         dicv:hasUnit unit:Euro ;
         dicv:hasTimeOfCreation "2021-05-25T12:00:00"^^xsd:dateTime ]
}
WHERE
{  :A1 dicv:hasProperty ?p . }
```

dicv-cq3 What are the constraints between properties? (e.g., less than, equal)

Example data:

```
INSERT DATA
{
 :A a dicp:Activity ;
    dice:occupiesTimeInterval [ dice:hasStart :AStart ;
                                dice:hasEnd :AEnd ] .
 :B a dicp:Activity ;
    dice:occupiesTimeInterval [ dice:hasStart :BStart ;
                                dice:hasEnd :BEnd ] .
 :C1 a dicv:BinaryConstraint ;
    dicv:hasComparison dicv:Less ;
    dicv:constrainsProperty1 :AStart ;
    dicv:constrainsProperty2 :AEnd .
   :C2 a dicv:BinaryConstraint ;
    dicv:hasComparison dicv:Less ;
    dicv:constrainsProperty1 :AEnd ;
    dicv:constrainsProperty2 :BStart .
   :C3 a dicv:BinaryConstraint ;
    dicv:hasComparison dicv:Less ;
    dicv:constrainsProperty1 :BStart ;
    dicv:constrainsProperty2 :BEnd .
}
```

12.1.3 Sample data for Digital Construction Entities (dice) ontology

Initialized by loading the Entities ontology:

```
LOAD <https://w3id.org/digitalconstruction/0.5/Entities/entities.ttl>
prefix dice: <https://w3id.org/digitalconstruction/0.5/Entities#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/XML/1998/namespace>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dica: <https://w3id.org/digitalconstruction/0.5/Agents#>
prefix : <http://example.com/id/>
```

dice-cq1 What entity has a given identifier in the given scope? (e.g., the room number in a building)

Example:

```
INSERT DATA
{
 :SGTIN96 a dice:Category ;
    rdfs:label "GTIN with a serial number indicating an instance of a trade item class" .
 :Device1 a dice:Device ;
    dice:isIdentifiedBy [a dice:Identifier ;
          dice:hasScope :SGTIN96 ;
          dice:hasLabel "012345.67890.10479832" ] .
 :B1Rooms a dice:EntityScope ;
    dice:hasEntitiesFrom :Building1 ;
    dice:hasLabel "Rooms" .
 :Space1 a dice:Location ;
    dice:isIdentifiedBy [ a dice:Identifier ;
                          dice:hasScope :B1Rooms ;
                          dice:hasLabel "101" ]
}
```

dice-cq2 What entities are classified in a given category in the given scope? (e.g., EG000819 in ETIM)

dice-cq3 What identifiers/categories does an entity have and in which scopes?

Example data:

```
INSERT DATA
{
 :GTIN a dice:Category ;
    rdfs:label "GS1 Trade Item Number indicating a class of trade items" .
  :ETIM a dice:Category ;
    rdfs:label "ETIM classification system of technical products".
 :Device1 a dice:BuildingObject ;
   dice:isClassifiedBy [ a dice:Category ;
      dice:hasScope :ETIM ;
      dice:hasLabel "EC000819" ] ,
    [ a dice:Category ;
      dice:hasScope :GTIN  ;
      dice:hasLabel "012345.67890" ].
 :Device2 a dice:BuildingObject ;
   dice:isClassifiedBy [ a dice:Category ;
      dice:hasScope :ETIM ;
      dice:hasLabel "EC000819" ].
}
```

dice-cq4 What type and instance have been assigned to an entity? (prescriptive, as in product selection)

Example data:

```
INSERT DATA
{
 :P1 a dice:Device ;
     dice:isAssignedToType [ a dice:Category ;
                             dice:hasLabel "9888001" ;
                             dice:hasScope :ETIM ]  ;
     dice:isAssignedToInstance [ a dice:Identifier ;
                                 dice:hasLabel "9888001" ;
                                 dice:hasScope :SGTIN96 ] .
}
```

dice-cq5 What entities are positioned in the given location?

Example data:

```
INSERT DATA {
 :Sensor1 a dice:Sensor ;
        dice:isLocatedIn :Loc1 .
 :Agent1 a dica:Agent ;
        dice:isLocatedIn :Loc1 . }
```

dice-cq6 What are the parts of a building object?

Example data:

```
INSERT DATA {
 :BO1 a dice:BuildingObject ;
     dice:hasContinuantPart :B02, :B03, :B04 .
 :B02 a dice:BuildingObject ;
     dice:hasContinuantPart :B021, :B022 .
 :B03 a dice:BuildingObject ;
     dice:hasContinuantPart :B031, :B032 , :B033 .
 :B04 a dice:BuildingObject . }
```

12.1.4 Sample data for Digital Construction Processes (dicp) ontology

Initialize by loading the Processes ontology:

```
LOAD <https://w3id.org/digitalconstruction/0.5/Processes/processes.ttl>
prefix dicp: <https://w3id.org/digitalconstruction/0.5/processes#>
prefix dice: <https://w3id.org/digitalconstruction/0.5/entities#>
prefix time: <http://www.w3.org/2006/time#>
prefix dicc: <https://w3id.org/digitalconstruction/0.5/contexts#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/xml/1998/namespace>
prefix xsd: <http://www.w3.org/2001/xmlschema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix ifc: <https://standards.buildingsmart.org/ifc/dev/ifc4/add2_tc1/owl>
prefix :    <http://example.com/dicp/id/>
```

dicp-cq1 What subactivities (or leaf-level subactivities) does the activity has?

Example data:

```
INSERT DATA
{
  :Renovation1 dicp:hasSubActivity :R1Design, :R1Procurement, :R1Construction, :R1Handover .
  :R1Design a dicp:Activity ; time:overlaps :R1Procurement .
  :R1Procurement a dicp:Activity ; time:overlaps :R1Construction .
  :R1Construction a dicp:Activity ; time:before :R1Handover .
  :R1Construction a dicp:Activity ;
                  dicp:hasSubActivity :R1ReplaceWindows , :R1InstallHeatPump .
```

```
   :R1ReplaceWindows a dicp:Activity ;
                          time:overlaps :R1InstallHeatPump  .
   :R1InstallHeatPump a dicp:Activity .
   :R1Handover a dicp:Activity
}
```

dicp-cq2 What entities (or input/output entities) is the given activity acting on?

dicp-cq3 In what location (or initial/final location) is the activity taking place?

dicp-cq4 What equipment are needed in the execution of an activity?

dicp-cq5 What resources have been assigned for the execution of an activity?

Example data:

```
INSERT DATA
{
  :Tool a dice:Category .
  :DrillingTool a dice:Category ;
     dice:hasLabel "Drill" ;
     dice:hasScope :Tool .
  :NailingTool a dice:Category ;
     dice:hasLabel "NailingTool" ;
     dice:hasScope :Tool .

  :ToolCapability a dice:Category .
  :WoodDrillingCapability a dice:Category ;
     dice:hasLabel "WoodDrilling" ;
     dice:hasScope :ToolCapability .
  :AutomaticNailingCapability a dice:Category ;
     dice:hasLabel "AutomaticNailing" ;
     dice:hasScope :AutomaticNailingCapability .

  :NailGun1 a dice:Equipment ;
           dice:isClassifiedBy :NailingTool ;
           dice:hasCapability :NailFasteningCapability .
  :SpiritLevel1 a dice:Equipment .
  :PaintRoller1 a dice:Equipment .
  :Drill1 a dice:Equipment ;
         dice:isClassifiedBy :DrillingTool ;
         dice:hasCapability :WoodDrillingCapability .
  :SurfaceGrinder1 a dice:Equipment .
  :DryWall1 a dice:BuildingObject , ifc:IfcWall .
  :Apartment101 a dice:ResidentialUnit .
  :Apartment102 a dice:ResidentialUnit .
   :DryWallConstruction1 a dicp:Activity ;
     dicp:hasOutputObject :DryWall1 ;
     dicp:hasLocation :Apartment101 ;
     dicp:hasSubActivity :EquipmentTransfer1, :FrameErection1, :BackboardInstallation1,
        :ElectricalWiring1, :FrontboardInstallation1, :WallLeveling1, :WallPainting1.
  :EquipmentTransfer1 a dicp:Activity ;
     dicp:hasInitialLocation :Apartment102 ;
     dicp:hasFinalLocation :Apartment101 ;
     dicp:hasObject :NailGun1, :SpiritLevel1, :Drill1, :SurfaceGrinder1 ;
     time:before :FrameErection1 .
  :FrameErection1 a dicp:Activity ;
     dicp:hasOutputObject :DryWall1 ;
     dicp:hasLocation :Apartment101 ;
```

```
    dicp:hasResourceRequirement [ a dicp:ResourceRequirement ;
                                  dicp:requiresResourceType :NailingTool ;
                        dicp:requiresResourceCapability :AutomaticNailingCapability ] ;
    dicp:hasAssignedResource [ a dicp:ResourceRole ; dice:isRoleOf :NailGun ] ;
    dicp:hasEquipment :NailGun1, :SpiritLevel1 ;
    time:before :BackboardInstallation1 .
  :BackboardInstallation1 a dicp:Activity ;
    dicp:hasObject :DryWall1 ;
    dicp:hasLocation :Apartment101 ;
    dicp:hasEquipment :Drill1, :SpiritLevel1 ;
    time:before :ElectricalWiring .
  :ElectricalWiring1 a dicp:Activity ;
    dicp:hasObject :DryWall1 ;
    dicp:hasLocation :Apartment101 ;
    dicp:hasResourceRequirement [ a dicp:ResourceRequirement ;
                                  dicp:requiresResourceType :DrillingTool ;
                        dicp:requiresResourceCapability :WoodDrillingCapability ] ;
    dicp:hasAssignedResource [ a dicp:ResourceRole ; dicp:isRoleOf :Drill1 ] ;
    dicp:hasEquipment :Drill1 ;
    time:before :FrontboardInstallation1 .
  :FrontboardInstallation1 a dicp:Activity ;
    dicp:hasObject :DryWall1 ;
    dicp:hasLocation :Apartment101 ;
    dicp:hasEquipment :Drill1, :SpiritLevel1 ;
    time:before :WallLeveling1 .
  :WallLeveling1 a dicp:Activity ;
    dicp:hasObject :DryWall1 ;
    dicp:hasLocation :Apartment101 ;
    dicp:hasEquipment :SurfaceGrinder1 ;
    time:before :WallPainting1.
  :WallPainting1 a dicp:Activity ;
    dicp:hasObject :DryWall1 ;
    dicp:hasEquipment :PaintRoller ;
    dicp:hasLocation :Apartment101 .
}
```

dicp-cq6 What is the time when the activity is executed? (planned and actual times)

Example data:

```
INSERT DATA
{
  :WeekPlan1 a dicc:Context ; dicc:hasContent :WeekPlan1Graph .
  GRAPH :WeekPlan1Graph {
    :FrameErection1 dice:occupiesTimeInterval
      [ dice:hasStart [ time:inXSDDateTime "2021-05-15T08:00:00"^^xsd:dateTime ] ;
          dice:hasEnd [ time:inXSDDateTime "2021-05-15T16:00:00"^^xsd:dateTime ] ]
  }
  :FrameErection1 dice:occupiesTimeInterval
    [ dice:hasStart [ time:inXSDDateTime "2021-05-16T09:00:00"^^xsd:dateTime ] ;
      dice:hasEnd [ time:inXSDDateTime "2021-05-17T12:00:00"^^xsd:dateTime ] ] }
```

12.1.5 Sample data for Digital Construction Agents (dica) ontology

Initialize by loading the Agents ontology:

```
LOAD <https://w3id.org/digitalconstruction/0.5/Agents/agents.ttl>
prefix dica: <https://w3id.org/digitalconstruction/0.5/agents#>
```

```
prefix dice: <https://w3id.org/digitalconstruction/0.5/entities#>
prefix dicp: <https://w3id.org/digitalconstruction/0.5/processes#>
prefix time: <http://www.w3.org/2006/time#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/xml/1998/namespace>
prefix xsd: <http://www.w3.org/2001/xmlschema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix :     <http://example.com/dica/id/>
```

dica-cq1 Who is the agent of an activity? (a person or organization)

Example data (continuing previous):

```
INSERT DATA
{
  :Carpenter1 a dica:Agent .
  :Electrician1 a dica:Agent .
  :Painter1 a dica:Agent .
  :FrameErection1 dica:hasAgent :Carpenter1 .
  :BackboardInstallation1 dica:hasAgent :Carpenter1 .
  :FrontboardInstallation1 dica:hasAgent :Carpenter1 .
  :ElectricalWiring1 dica:hasAgent :Electrician1.
  :WallLeveling dica:hasAgent :Painter1 .
  :WallPainting dica:hasAgent :Painter1 }
```

dica-cq2 What is the consortium of the given renovation project?

dica-cq3 Who is leading the consortium of the given renovation project?

Example data (continuing previous):

```
INSERT DATA {
  :Owner1 a dica:Agent .
  :ConstructionManager1 a dica:Corporation .
  :Architect1 a dica:Corporation .
  :Contractor1 a dica:Corporation .
  :Appointment1 a dica:Appointment ;
              dica:hasAppointingParty :Owner1 ;
              dica:hasAppointedParty :ConstructionManager1 .
  :Appointment2 a dica:Appointment ;
              dica:hasAppointingParty :ConstructionManager1 ;
              dica:hasAppointedParty :Architect1 .
  :Appointment3 a dica:Appointment ;
              dica:hasAppointingParty :ConstructionManager1 ;
              dica:hasAppointedParty :Contractor1 .
  :ProjectTeam1 a dica:ProjectTeam ;
              dice:hasMember :ConstructionManager1, :Architect1, :Contractor1 ;
              dica:hasLeadAppointedParty :ConstructionManager1 .
  :Project1 a dicp:Project ;
          dica:hasExecutingAgent :ProjectTeam1 }
```

dica-cq4 Who are the stakeholders (owners and occupants) related to an activity?

Example data:

```
INSERT DATA
{
  :ResidentialOwner1 a dica:LegallyCompetentNaturalPerson ;
                dice:isOwnerOf :Apartment101 .
  :Occupant1 a dica:Person ;
          dice:isOccupantIn :Apartment101 .
```

```
  :Apartment101 a dice:ResidentialUnit .
  :Activity1 dicp:hasLocation :Apartment101 }
```

12.1.6 Sample data on Digital Construction Information (dici) ontology

Initialize by loading the Information ontology:

```
load <https://w3id.org/digitalconstruction/0.5/information/information.ttl>
prefix dicp: <https://w3id.org/digitalconstruction/0.5/processes#>
prefix dica: <https://w3id.org/digitalconstruction/0.5/agents#>
prefix dice: <https://w3id.org/digitalconstruction/0.5/entities#>
prefix dicc: <https://w3id.org/digitalconstruction/0.5/contexts#>
prefix dici: <https://w3id.org/digitalconstruction/0.5/information#>
prefix time: <http://www.w3.org/2006/time#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/xml/1998/namespace>
prefix xsd: <http://www.w3.org/2001/xmlschema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix :     <http://example.com/dici/id/>
```

dici-cq1 What is the information model of the given renovation project? (according to ISO 19650)

Example data:

```
INSERT DATA {
  :Project1 a dicp:RenovationProject .

  :PIM1 a dici:ProjectInformationModel ;
       dici:isAbout :Project1 }
```

dici-cq2 What information containers are active in the current state? (contain current information)

Example data:

```
INSERT DATA
{
  :Project1 a dicp:RenovationProject .
    :PIM1 dici:includesContainer :ProjectExecution ;
       dici:isAbout :Project1 ;
       dici:includesModel :ArchitecturalContainers1, :StructuralContainers1 ;
       dici:hasDefaultContainerSet :DefaultContainers .
    :ArchitecturalContainers1 a dici:InformationModel ;
      dici:includesContainer :ArchLOD300, :ArchLOD350 .
  :StructuralContainers1 a dici:InformationModel ;
       dici:includesContainer :StructLOD300, :StructLOD350 .
  :DefaultContainers a dicc:ContextSet ;
       dicc:hasActiveContext :ProjectExecution, :ArchLOD350, :StructLOD350 }
```

dici-cq3 What information (or output information) does the given activity act on? (information flow)

Example data:

```
INSERT DATA
{
  :QuantityEstimation1 a dicp:Activity ;
     dici:hasInformation :QuantityEstimationGuide ;
     dici:hasInputInformation :ArchBIM1, :StructBIM1, :MEPBIM1 ;
     dici:hasOutputInformation :QuantityTakeOff1 .
  :QuantityTakeOff1 a dici:InformationContentEntity .
  :QuantityEstimationGuide a dici:InformationContentEntity .
  :ArchBIM1 a dici:BuildingInformationModel .
  :StructBIM1 a dici:BuildingInformationModel .
```

:MEPBIM1 a dici:BuildingInformationModel }

dici-cq4 Who produced the specific information content and when? (metadata)

Example data:

```
INSERT DATA
{
  :QuantityEstimation1 dici:isCreatedBy :ProjectManager1 ;
                   dici:isCreatedAt "2021-05-15T12:00:00"^^xsd:dateTime
}
```

12.1.7 Sample data on Digital Construction Materials (dicm) ontology

The prefixes used for the example data are listed as follows:

```
load < https://digitalconstruction.github.io/Materials/v/0.5/materials.ttl>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/xml/1998/namespace>
prefix xsd: <http://www.w3.org/2001/xmlschema#>
prefix dice: <https://w3id.org/digitalconstruction/0.5/entities#>
prefix dicm: <https://w3id.org/digitalconstruction/0.5/materials#>
prefix dicu: <https://w3id.org/digitalconstruction/0.5/units#>
prefix dicv: <https://w3id.org/digitalconstruction/0.5/variables#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix :     <http://example.com/id/>
```

dicm-cq1 How the material object structure is defined?

dicm-cq2 What are the layers of an BuildingObject ?

dicm-cq3 What is the adjacent layer of a layer ?

dicm-cq4 What is the material of the layer or building object?

dicm-cq5 How different materials are classified?

Example Data:

```
INSERT DATA
{
:Wall1 a dice:BuildingObject;
                         dicm:hasLayerSet :Layerset1 .
:Layerset1 dicm:hasLayer :Layer1,
                                      :Layer2,
                                      :Layer3 .
:Layer1 dicm:hasAdjacentLayer :Layer2 .
:Layer2 dicm:hasAdjacentLayer :Layer3 .
:Layer2 dicm:hasAdjacentLayer :Layer1 .
:Layer1 dicm:hasMaterial :Material1 .
:Layer2 dicm:hasMaterial :Material2 .
:Layer3 dicm:hasMaterial :Material3 .
:Material1 a dicm:InorganicNonMetallicMaterial .
:Material2 a dicm:CompositeMaterial .
:Material3 a dicm:InorganicNonMetallicMaterial .
}
```

dicm-cq6 How the material properties are defined?

Example Data:

###Basic representation: a simple data property ###

```
INSERT DATA
{
:Material1 dicm:hasThermalConductivity "0.65"^^xsd:double ;
                    dice:hasLabel "Gypsum Wall Board"^^xsd:string .
:Material2 dicm:hasThermalConductivity "0.54"^^xsd:double ;
                    dice:hasLabel "Brick"^^xsd:string .
:Material3 dicm:hasThermalConductivity "0.51"^^xsd:double ;
                    dice:hasLabel "Plaster"^^xsd:string .
}
```

dicm-cq7 How the material properties are objectified?

Example Data:

```
###Complex representation: an objectified property###
INSERT DATA
{
:Material1 rdf:type owl:NamedIndividual ;
          dicv:hasProperty :property1 .
:Material2 rdf:type owl:NamedIndividual ;
          dicv:hasProperty :property2 .
:Material3 rdf:type owl:NamedIndividual ;
          dicv:hasProperty :property3 .
:property1 rdf:type owl:NamedIndividual ;
                    dicv:hasUnit dicu:W_PER_m-K ;
                    dicv:isPropertyFor dicm:hasThermalConductivity ;
                    dicv:hasValue "0.65"^^xsd:double .
:property2 rdf:type owl:NamedIndividual ;
                    dicv:hasUnit dicu:W_PER_m-K ;
                    dicv:isPropertyFor dicm:hasThermalConductivity ;
                    dicv:hasValue "0.54"^^xsd:double .
:property3 rdf:type owl:NamedIndividual ;
                    dicv:hasUnit dicu:W_PER_m-K ;
                    dicv:isPropertyFor dicm:hasThermalConductivity ;
dicv:hasValue "0.51"^^xsd:double .

}
```

12.1.8 Sample data for Digital Construction Occupancy (dicob) ontology

Initialized by loading the Occupancy ontology:

```
load <https://w3id.org/digitalconstruction/0.5/occupancy/occupancy.ttl>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/xml/1998/namespace>
prefix xsd: <http://www.w3.org/2001/xmlschema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dice: <https://w3id.org/digitalconstruction/0.5/entities#>
prefix dica: <https://w3id.org/digitalconstruction/0.5/agents#>
prefix dicob: <https://w3id.org/digitalconstruction/0.5/occupancy#>
prefix unit: <http://qudt.org/2.1/vocab/unit/>
prefix dicv: <https://w3id.org/digitalconstruction/0.5/variables#>
prefix time: <http://www.w3.org/2006/time#>
prefix :     <http://example.com/id/>
```

dicob-cq1 What is the number of occupants in the apartment?

Example data:

```
INSERT DATA
{
 :Apartment101 a dice:ResidentialUnit .
 :Apartment102 a dice:ResidentialUnit .
 :Mary dica:isOccupantIn :Apartment101 .
 :Joe dica:isOccupantIn  :Apartment101 .
 :Bill dica:isOccupantIn :Apartment102 }
```

dicob-cq2 What are the occupancy schedules performed by the occupants in the building environment?

Example data:

```
INSERT DATA
{
 :Cooking1 a dicob:OccupancyActivity ;
    dica:hasAgent :Mary ;
    dice:occupiesTimeInterval
        [ dice:hasStart [ time:inXSDDateTime "2021-05-15T12:00:00"^^xsd:dateTime ] ;
               dice:hasEnd [ time:inXSDDateTime "2021-05-15T12:40:00"^^xsd:dateTime ] ] .
 :Vacuuming1 a dicob:OccupancyActivity ;
    dica:hasAgent :Mary ;
    dice:occupiesTimeInterval
       [ dice:hasStart [ time:inXSDDateTime "2021-05-15T15:00:00"^^xsd:dateTime ] ;
              dice:hasEnd [ time:inXSDDateTime "2021-05-15T15:30:00"^^xsd:dateTime ] ] .
 :PianoPlaying a dicob:OccupancyActivity ;
    dica:hasAgent :Joe ;
    dice:occupiesTimeInterval
        [ dice:hasStart [ time:inXSDDateTime "2021-05-15T18:00:00"^^xsd:dateTime ] ;
               dice:hasEnd [ time:inXSDDateTime "2021-05-15T20:00:00"^^xsd:dateTime ] ] }
```

dicob-cq3 What is the birthyear of the occupants placed in a residential apartment?

Example data:

```
INSERT DATA
{
 :Mary dica:hasBirthYear "1992" .
 :Joe  dica:hasBirthYear "1982" .
 :Bill dica:hasBirthYear "1972" }
```

dicob-cq4 and cq5 What are the minimum/maximum indoor environmental quality (temperature, luminance, noise) in the apartment?

Example data:

```
INSERT DATA
{
 :Apartment101 dicv:hasProperty
     [ a dicob:Temperature ;
          dicv:hasUnit unit:DEG_C ;
       dicv:hasPropertyState
          [ a dicv:QuantitativeState ;
            dicv:timeOfCreation "2021-05-15T12:00:00"^^xsd:dateTime ;
            dicv:hasValue "18"^^xsd:integer ] ,
          [ a dicv:QuantitativeState ;
            dicv:timeOfCreation "2021-05-16T12:00:00"^^xsd:dateTime ;
            dicv:hasValue "22"^^xsd:integer ] ,
          [ a dicv:QuantitativeState ;
            dicv:timeOfCreation "2021-05-17T12:00:00"^^xsd:dateTime ;
            dicv:hasValue "20"^^xsd:integer ] ] }
```

dicob-cq6, cq7, cq8 Which is the temperature/noise/luminance sensor in building zone?

Example data:

```
INSERT DATA
{
  :TemperatureSensor1 a dicob:TemperatureSensor ;
                    dice:isLocatedIn :Apartment101 .
  :NoiseSensor1 a dicob:NoiseSensor ;
              dice:isLocatedIn :Apartment102 .
  :LuminanceSensor1 a dicob:LuminanceSensor ;
                  dice:isLocatedIn :Apartment102  }
```

12.1.9 Sample data for Digital Construction Energy (dices) ontology

Initialized by loading the Energy ontology:

```
load <https://w3id.org/digitalconstruction/0.5/energy.ttl>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xml: <http://www.w3.org/xml/1998/namespace>
prefix xsd: <http://www.w3.org/2001/xmlschema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dice: <https://w3id.org/digitalconstruction/0.5/entities#>
prefix dices: <https://w3id.org/digitalconstruction/0.5/energy#>
prefix dicv: <https://w3id.org/digitalconstruction/0.5/variables#>
prefix dici: <https://w3id.org/digitalconstruction/0.5/information#>
prefix :     <http://example.com/dices/id/>
```

dices-cq1 and cq2 Which is the generation/storage system in building zone?

Example data:

```
INSERT DATA
{
  :Generator1 a dices:Generator ;
            dice:isLocatedIn :BuildingUnit1 .
  :Battery1 a dices:BatteryStorage ;
          dice:isLocatedIn :Apartment102 }
```

dices-cq3 and cq4 What is the operational service for the HVAC_1/Photovoltics_1 device?

Example data:

```
INSERT DATA
{
  :HVAC_1 a dices:HVACDevice ;
        dices:offersService :ComfortManagement1, :ServiceAggregation1  .
  :ComfortManagement1 a dices:ComformManagement .
  :ServiceAggregation1 a dices:ServiceAggregation .
  :Photovotaics_1 a dices:Photovotaics;
              dices:offersService :SelfConsumptionOptimiation1 .
  :SelfConsumptionOptimiation1 a dices:SelfConsumptionOptimization }
```

dices-cq5 What are the LCA values of the Photovoltaics_1?

dices-cq6 What are the LCC values of the Photovoltaics_1?

Example data:

```
INSERT DATA
{
  :Photovoltaics_1 a dices:Photovotaics .
```

```
 :LifeCycleAssessment a dices:LifeCycleAssessment;
     dici:isAbout :Photovoltaics_1 ;
     dici:hasLifeCycleImpact [ a dices:OzoneDepletionPotential ;
                               dicv:hasValue "0.01"^^xsd:decimal ] ,
                          [ a dici:HumanToxicityPotential ;
                           dicv:hasValue "2.26e-03"^^xsd:double ] ;
     dici:documentsProperty [ a dices:CapitalCost ;
                               dicv:hasValue "2000" ] ,
                          [ a dices:EngineeringCost ;
                               dicv:hasValue "1000" ] ,
                          [ a dices:MaintenanceCost ;
                               dicv:hasValue "1200" ] }
```

dices-cq7 Which is the meter system in building zone?

dices-cq8 What is the primary energy conversion factor for the meter system?

dices-cq9 What is the CO2 emission conversion factor for the meter system?

Example data:

```
INSERT DATA
{
 :Meter1 a dices:Meter ;
       dice:isLocatedIn :Apartment101 ;
 dices:hasProperty [ a dices:FossilFuelPotential ;
                    dicv:hasValue "0.4"^^xsd:decimal ] ,
              [ a dices:PrimaryEnergyFactor ;
                    dicv:hasValue "0.3"^^xsd:decimal ] }
```

12.1.10 Sample data for Digital Construction Lifecycle (dicl) ontology

The prefixes used for the dicl ontology are listed as follows:

```
Load: < https://digitalconstruction.github.io/Lifecycle/v/0.5/lifecycle.ttl>
PREFIX : <http://example.com/id#>
PREFIX dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX diclvl: <https://w3id.org/digitalconstruction/0.5/Levels#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX obda: <https://w3id.org/obda/vocabulary#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dice: <https://w3id.org/digitalconstruction/0.5/Entities#>
PREFIX dicv: <https://w3id.org/digitalconstruction/0.5/Variables#>
PREFIX dicp: <https://w3id.org/digitalconstruction/0.5/Processes#>
PREFIX dica: <https://w3id.org/digitalconstruction/0.5/Agents#>
PREFIX dicstg: <https://w3id.org/digitalconstruction/0.5/Stages#>
```

dicl-cq1: How can the BIM data representation be adjusted or modified to different LOD systems?

dicl-cq2: what is the link between the lod system and its levels?

dicl-cq3: what is the relation between the lod classification-level and the lod scale?

dicl-cq4: what is the relation between lod scales?

dicl-cq5: how is the bim object is represented?

dicl-cq6: how to represent multiple versions of information about the same object?

dicl-cq7: how the object properties and values for a specific lod level are defined?

dicl-cq8: what are the sources for lod data?

dicl-cq9: how the activities are defined in the renovation workflow?

dicl-cq10: how to identify the sequence of activities?

dicl-cq11: how are the stakeholders or agents related to activities?

dicl-cq12: how does the renovation process linked to bim data represented in lod-sensitive manner?

dicl-cq13: how to represent specific use cases within the renovation workflow?

dicl-cq14: how to enable the representation of multiple bls systems and/or stages?

dicl-cq15: what is the link between the bls system and its respective stages?

dicl-cq16: what is the relation between stages?

dicl-cq17: what is the relation between the main-stage to sub-stage of other main-stage?

dicl-cq18: how to represent the mapping between different bls systems?

Example Data:

```
INSERT DATA
{
        dicl:hasLevel rdf:type owl:ObjectProperty ;
        rdfs:domain dicl:LODFramework ;
        rdfs:range dicl:LODLevel ;
        owl:propertyChainAxiom ( dicl:hasLevel
                                 dicl:hasSubLevel
                               ) .
  dicl:hasNextLevel rdf:type owl:ObjectProperty ;
            owl:inverseOf dicl:hasPreviousLevel ;
            rdf:type owl:TransitiveProperty ;
            rdfs:domain dicl:LODLevel ;
            rdfs:range dicl:LODLevel .
  dicl:hasPreviousLevel rdf:type owl:ObjectProperty ,
                         owl:TransitiveProperty ;
              rdfs:domain dicl:LODLevel ;
              rdfs:range dicl:LODLevel .
  dicl:hasSubLevel rdf:type owl:ObjectProperty ;
           owl:inverseOf dicl:hasSuperLevel ;
           rdf:type owl:TransitiveProperty ;
           rdfs:domain dicl:LODLevel ;
           rdfs:range dicl:LODLevel .
  dicl:hasSuperLevel rdf:type owl:ObjectProperty ,
                      owl:TransitiveProperty ;
            rdfs:domain dicl:LODLevel ;
            rdfs:range dicl:LODLevel .
  dicl:LODFramework rdf:type owl:Class .
  dicl:LODLevel rdf:type owl:Class .
  diclvl:AsBuilt rdf:type owl:NamedIndividual ;
        dicl:hasSubLevel diclvl:LOD_500 .
        diclvl:AsDesigned rdf:type owl:NamedIndividual ;
          dicl:hasNextLevel diclvl:AsBuilt ;
```

```
    dicl:hasSubLevel diclvl:LOD_100 ,
                     diclvl:LOD_200 ,
                     diclvl:LOD_300 ,
                     diclvl:LOD_350 ,
                     diclvl:LOD_400 .
   diclvl:LOD_100 rdf:type owl:NamedIndividual ;
   dicl:hasNextLevel diclvl:LOD_200 ;
   rdfs:label "LOD 100"^^xsd:string .
   diclvl:LOD_200 rdf:type owl:NamedIndividual ;
   dicl:hasNextLevel diclvl:LOD_300 ;
   rdfs:label "LOD 200"^^xsd:string .
   diclvl:LOD_300 rdf:type owl:NamedIndividual ;
   dicl:hasNextLevel diclvl:LOD_350 ;
   rdfs:label "LOD 300"^^xsd:string .
   diclvl:LOD_350 rdf:type owl:NamedIndividual ;
   dicl:hasNextLevel diclvl:LOD_400 ;
   rdfs:label "LOD 350"^^xsd:string .
   diclvl:LOD_400 rdf:type owl:NamedIndividual ;
   dicl:hasNextLevel diclvl:LOD_500 ;
   rdfs:label "LOD 400"^^xsd:string .
   diclvl:LOD_500 rdf:type owl:NamedIndividual ;
   rdfs:label "LOD 500"^^xsd:string .
   diclvl:USA_BIMForum rdf:type owl:NamedIndividual ;
      dicl:hasLevel diclvl:AsBuilt ,
                    diclvl:AsDesigned ;
      rdfs:label "USA BIMForum"^^xsd:string .
dicl:hasNextStage rdf:type owl:ObjectProperty ;
          owl:inverseOf dicl:hasPreviousStage ;
          rdf:type owl:TransitiveProperty ;
          rdfs:domain dicl:BLStage ;
          rdfs:range dicl:BLStage ;
          owl:propertyChainAxiom ( dicl:hasNextStage
                                   dicl:hasSubStage
                                 ) .
dicl:hasPreviousStage rdf:type owl:ObjectProperty ,
                      owl:TransitiveProperty ;
             rdfs:domain dicl:BLStage ;
             rdfs:range dicl:BLStage ;
             owl:propertyChainAxiom ( dicl:hasPreviousStage
                                      dicl:hasSubStage
                                    ) .
dicl:hasStage rdf:type owl:ObjectProperty ;
       rdfs:domain dicl:BLSFramework ;
       rdfs:range dicl:BLStage ;
       owl:propertyChainAxiom ( dicl:hasStage
                                dicl:hasSubStage
                              ) .
dicl:hasSubStage rdf:type owl:ObjectProperty ;
         owl:inverseOf dicl:hasSuperStage ;
         rdf:type owl:TransitiveProperty ;
         rdfs:domain dicl:BLStage ;
         rdfs:range dicl:BLStage .
dicl:hasSuperStage rdf:type owl:ObjectProperty ,
                   owl:TransitiveProperty ;
          rdfs:domain dicl:BLStage ;
          rdfs:range dicl:BLStage .
```

```
dicl:isRelaventWith rdf:type owl:ObjectProperty ,
                    owl:SymmetricProperty ;
        rdfs:domain dicl:BLStage ;
        rdfs:range dicl:BLStage .
dicstg:BLSFramework rdf:type owl:Class .
dicstg:BLStage rdf:type owl:Class .
dicstg:BS_EN_16310 rdf:type owl:NamedIndividual ;
      dicl:hasStage dicstg:0.BS_EN_Initiative ,
             dicstg:1.BS_EN_Initiation ,
             dicstg:2.BS_EN_Design ,
             dicstg:3.BS_EN_Procurement ,
             dicstg:4.BS_EN_Construction ,
             dicstg:5.BS_EN_Use ,
             dicstg:6.BS_EN_End_of_Life .
dicstg:0.BS_EN_Initiative rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:1.BS_EN_Initiation ;
                              dicl:hasSubStage dicstg:0.1.BS_EN_Market_study ,
                                       dicstg:0.2.BS_EN_Business_case .
    dicstg:0.1.BS_EN_Market_study rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:0.2.BS_EN_Business_case .
    dicstg:0.2.BS_EN_Business_case rdf:type owl:NamedIndividual .
    dicstg:1.BS_EN_Initiation rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:2.BS_EN_Design ;
                              dicl:hasSubStage dicstg:1.1.BS_EN_Project_initiation ,
                                       dicstg:1.2.BS_EN_Feasibility_study ,
                                       dicstg:1.3.BS_EN_Project_definition .
dicstg:1.1.BS_EN_Project_initiation rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:1.2.BS_EN_Feasibility_study .
    dicstg:1.2.BS_EN_Feasibility_study rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:1.3.BS_EN_Project_definition .
    dicstg:1.3.BS_EN_Project_definition rdf:type owl:NamedIndividual .
    dicstg:2.BS_EN_Design rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:3.BS_EN_Procurement ;
                              dicl:hasSubStage dicstg:2.1.BS_EN_Conceptual_design ,
                                       dicstg:2.2.BS_EN_Preliminary_design ,
                                       dicstg:2.3.BS_EN_Developed_design ,
                                       dicstg:2.4.BS_EN_Technical_design ,
                                       dicstg:2.5.BS_EN_Detailed_design .
dicstg:2.1.BS_EN_Conceptual_design rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:2.2.BS_EN_Preliminary_design .

dicstg:2.2.BS_EN_Preliminary_design rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:2.3.BS_EN_Developed_design .

dicstg:2.3.BS_EN_Developed_design rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:2.4.BS_EN_Technical_design .
dicstg:2.4.BS_EN_Technical_design rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:2.5.BS_EN_Detailed_design .
dicstg:2.5.BS_EN_Detailed_design rdf:type owl:NamedIndividual .
dicstg:3.BS_EN_Procurement rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:4.BS_EN_Construction ;
                              dicl:hasSubStage dicstg:3.1.BS_EN_Procurement ,
                                       dicstg:3.2.BS_EN_Construction_contracting .
dicstg:3.1.BS_EN_Procurement rdf:type owl:NamedIndividual ;
                              dicl:hasNextStage dicstg:3.2.BS_EN_Construction_contracting .
dicstg:3.2.BS_EN_Construction_contracting rdf:type owl:NamedIndividual .
```

```
dicstg:4.BS_EN_Construction rdf:type owl:NamedIndividual ;
                                            dicl:hasNextStage dicstg:5.BS_EN_Use ;
                                            dicl:hasSubStage dicstg:4.1.BS_EN_Pre_construction ,
                                                      dicstg:4.2.BS_EN_Construction ,
                                                      dicstg:4.3.BS_EN_Commissioning ,
                                                      dicstg:4.4.BS_EN_Handover ,
                                                     dicstg:4.5.BS_EN_Regulatory_approval .
dicstg:4.1.BS_EN_Pre_construction rdf:type owl:NamedIndividual ;
                                               dicl:hasNextStage dicstg:4.2.BS_EN_Construction .
dicstg:4.2.BS_EN_Construction rdf:type owl:NamedIndividual ;
                                             dicl:hasNextStage dicstg:4.3.BS_EN_Commissioning .
dicstg:4.3.BS_EN_Commissioning rdf:type owl:NamedIndividual ;
                                             dicl:hasNextStage dicstg:4.4.BS_EN_Handover .
dicstg:4.4.BS_EN_Handover rdf:type owl:NamedIndividual ;
                                           dicl:hasNextStage dicstg:4.5.BS_EN_Regulatory_approval .
dicstg:4.5.BS_EN_Regulatory_approval rdf:type owl:NamedIndividual .
dicstg:5.BS_EN_Use rdf:type owl:NamedIndividual ;
                                       dicl:hasNextStage dicstg:6.BS_EN_End_of_Life ;
                                       dicl:hasSubStage dicstg:5.1.BS_EN_Operation ,
                                                 dicstg:5.2.BS_EN_Maintenance .
dicstg:5.1.BS_EN_Operation rdf:type owl:NamedIndividual ;
                                            dicl:hasNextStage dicstg:5.2.BS_EN_Maintenance .
dicstg:5.2.BS_EN_Maintenance rdf:type owl:NamedIndividual .
dicstg:6.BS_EN_End_of_Life rdf:type owl:NamedIndividual ;
                                            dicl:hasSubStage dicstg:6.1.BS_EN_Revamping ,
                                                      dicstg:6.2.BS_EN_Dismantling .
dicstg:6.1.BS_EN_Revamping rdf:type owl:NamedIndividual ;
                                            dicl:hasNextStage dicstg:6.2.BS_EN_Dismantling .
dicstg:6.2.BS_EN_Dismantling rdf:type owl:NamedIndividual .
dicstg:ISO_22263 rdf:type owl:NamedIndividual ;
        dicl:hasStage dicstg:ISO_Design .
dicstg:ISO_Design dicl:isRelaventWith dicstg:2.BS_EN_Design .
:GlobalId_2af9a9d4-6443-4ac2-b74d-e584c11f4652
                                     a                    dice:BuildingObject;
                                     rdfs:label           "Basic Wall:500+100:2145690";
                                     dicl:hasID           "0g_QdKP4DAmhTDvOJ17qPI";
                                     dicv:hasProperty     :IfcPropertySingleValue_2024,
                                                :IfcPropertySingleValue_2033 .

      :IfcPropertySingleValue_2024
                         a                          dicv:Property;
                         rdfs:label              "ThermalTransmittance";
                         dicv:hasPropertyState         :IfcPropertySingleValue_2024_LOD300 ;
                         dicv:hasPropertyState         :IfcPropertySingleValue_2024_LOD200 .
      :IfcPropertySingleValue_2024_LOD300
                         a                   dicv:PropertyState;
                         dicl:hasLODLevel           diclvl:LOD_300;
                         dicl:hasPropertyName       "ThermalTransmittance";
                         dicv:hasValue              "1.087"^^xsd:double ;
                         dicl:isDerivedFrom         :Polish_site_LOD300_model .
      :IfcPropertySingleValue_2024_LOD200
                         a                   dicv:PropertyState;
                         dicl:hasLODLevel           diclvl:LOD_200;
                         dicl:hasPropertyName       "ThermalTransmittance";
                         dicv:hasValue              "1.717"^^xsd:double ;
                         dicl:isDerivedFrom         :Polish_site_LOD200_model .
```

```
:IfcPropertySingleValue_2033
                    a                           dicv:Property;
                    rdfs:label               "Width";
                    dicv:hasPropertyState          :IfcPropertySingleValue_2033_LOD300 ;
                    dicv:hasPropertyState          :IfcPropertySingleValue_2033_LOD200 .

:IfcPropertySingleValue_2033_LOD300
                    a                      dicv:PropertyState;
                    dicl:hasLODLevel             diclvl:LOD_300;
                    dicl:hasPropertyName         "Width";
                    dicv:hasValue                "600"^^xsd:double ;
                    dicl:isDerivedFrom           :Polish_site_LOD300_model .

:IfcPropertySingleValue_2033_LOD200
                    a                      dicv:PropertyState;
                    dicl:hasLODLevel             diclvl:LOD_200;
                    dicl:hasPropertyName         "Width";
                    dicv:hasValue                "640"^^xsd:double ;
                    dicl:isDerivedFrom           :Polish_site_LOD200_model .

diclvl:LOD_300          a                         dicl:LODLevel;
                    rdfs:label                   "LOD 300" .
diclvl:LOD_200          a                         dicl:LODLevel;
                    rdfs:label                   "LOD 200" .
dicstg:3.5_DetailedDesign              rdf:type                    dicl:BLStage ;
                        dicl:hasActivity         :125_BuildingPermissions ,
                                                                   :124_TechnicalDetails ,
                                                                   :123_ProductionOfPlans .

:125_BuildingPermissions       rdf:type                    dicp:Activity ;
              dica:hasAgent             :Project_Leader ;
              dicp:hasObject           :GlobalId_2af9a9d4-6443-4ac2-b74d-e584c11f4652 .

:Project_Leader              rdf:type                    dica:Agent ;
              dicl:consumesFrom   :125_BuildingPermissions .
:Architectural_Designer        rdf:type                    dica:Agent ;
              dicl:providesTo      :125_BuildingPermissions .
:Building_Services_Designer    rdf:type                    dica:Agent ;
              dicl:providesTo      :125_BuildingPermissions ;
              dicl:processFrom      :123_ProductionOfPlans ;
              dicl:consumesFrom     :123_ProductionOfPlans  .
:123_ProductionOfPlans         rdf:type                    dicp:Activity ;
              dica:hasAgent             :Architectural_Designer ;
              dicp:hasObject            :GlobalId_2af9a9d4-6443-4ac2-b74d-e584c11f4652 .
:124_TechnicalDetails          rdf:type                    dicp:Activity ;
              dica:hasAgent             :Building_Services_Designer ;
              dicp:hasObject            :GlobalId_2af9a9d4-6443-4ac2-b74d-e584c11f4652 .
:UsecaseEM9                      rdf:type                    dicl:InformationalUsecase ;
                                 rdfs:label  "Prepare detailed design" ;
                                 dicl:hasRepresents         :123_ProductionOfPlans ,
                                                            :124_TechnicalDetails ,
                                                            :125_BuildingPermissions .
}
```