BIM4EEB

BIM based fast toolkit for
Efficient rEnovation in Buildings

D3.3 An ontology for representing data and information stored in BIM models at different LOD corresponding to renovation process modelling

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

D3.3 An ontology for representing data and information stored in BIM models at different LOD corresponding to renovation process modelling

Work Package	WP3
Lead Partner	University College Cork (UCC)
Contributing Partner(s)	PoliMi, VTT, SUITE5, OneTeam, TUD, VisuaLynk
Dissemination Level	Public
Type	Report
Due date	31/08/2019
Updated version 3.0 submission date	07/02/2022
Version	3.0

DOCUMENT HISTORY

Version	Date	Comments	Main Authors
0.1	23.04.2019	ToC created	K.Menzel, J.Karlapudi, P.Valluru (TUD)
0.2	25.04.2019	Draft version presented for review and feedback	A.Hryshchenko (UCC)
0.3	07.05.2019	TOC updated with comments from WPL	K.Menzel, J.Karlapudi, P.Valluru (TUD)
0.4	15.07.2019	Draft version populated with comments from meeting in Helsinki for the first review	K.Menzel, J.Karlapudi, P.Valluru (TUD), A.Hryshchenko (UCC)
0.5	31.7.2019	Updates wrt. BLS, product-process matrix, LOD matrix	K.Menzel, J.Karlapudi, P.Valluru (TUD), A.Hryshchenko (UCC)
0.5.1	15.08.2019	BIM product modelling, use case definitions and ontology structure included	K.Menzel, J.Karlapudi, P.Valluru (TUD), A.Hryshchenko (UCC)
0.6	22.08.2019	Included ontology requirements for LOD support	S. Törmä (VisuaLynk)
0.7	23.28.2019	Version for the internal review	A.Hryshchenko (UCC)
0.8	26.08.2019	Comments received; changes being integrated	A.Hryshchenko (UCC)
0.9	29.08.2019	Final corrections integrated	A.Hryshchenko (UCC)
1.0	30.08.2019	Final version for submission	WP3 Partners
1.1	31.08.2019	Updates by WPL	K.Menzel (TUD)
1.2	26.06.2020	Additional Annex II was attached	J.Karlapudi (TUD)
1.3	23.07.2020	Comments from Karsten are updated	K.Menzel, P.Valluru (TUD)
1.4	24.07.2020	Final Version for submission	J.Karlapudi (TUD)
1.5	04.05.2021	Updates to the document based on Review report	J.Karlpudi (TUD), A.Hryshchenko (UCC)
1.6	24.05.2021	Internal review comments from the Partners	VTT, OneTeam
1.7	31.05.2021	Updated version based on Partners comments	J.Karlapudi (TUD)
1.8	02.06.2021	Review of PoliMi' team, and plagiarism checks	B.Daniotti, M.Signorini (PoliMi), T.Toniatti, (FPM/PoliMi)
1.9	02.06.2021	Updates and adjustments based on suggestions	J.Karlapudi (TUD)
2.0	03.06.2021	PC approval and submission	B.Daniotti (PoliMi)
2.1	28.06.2021	Final updates by WP-Leader	K.Menzel (TUD)
2.2	28.06.2021	All markup an track changes accepts	K.Menzel (TUD)
2.3	30.11.2021	Updates and adjustments based on reviewer suggestions	J.Karlapudi (TUD)
2.4	09.01.2022	Internal review comments from the Partners and WP Leader	S.Törmä (VisuaLynk), T.Vesanen (VTT), K.Menzel (TUD)
2.5	21.01.2022	Updates based on partner and WP leader suggestions	J.Karlapudi (TUD)
2.6	28.01.2022	Final check	S.Törmä (VisuaLynk), K.Menzel (TUD)

2.7	01.02.2022	Submission to Project Coordinator (WP Leader's approval)	J.Karlapudi (TUD)
3.0 FINAL	02.02.2022	Final version, 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 clearly indicated otherwise. Acknowledgement 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

Building Information Modelling (BIM) has the potential to become a technology that will help to use a holistic information repository to generate and represent relevant information in different stages of the building lifecycle to dedicated groups of stakeholders. Substantial efforts were made in the past to develop open, well-documented meta-data models and complement dictionaries to ensure integrated, semantically harmonised data management. The buildingSmart Consortium supports the development and maintenance of one of the most prominent openBIM meta-data models currently available, called Industry Foundation Classes (IFC).

The IFC-meta data model is very comprehensive. Building elements and building processes can be described using numerous additional classes and property sets. However, there is a need to clearly specify when (i.e. at what stage of a building's lifecycle), by whom (i.e. stakeholder or role) and what level such information must be added to a BIM model. To specify this, two distinct concepts were developed and defined by the AECO community, such as Levels of Detail (LOD) and Building Lifecycle Stages.

The concept of Levels of Detail (LOD) represents the scope and completeness of the information that is required for a building's documentation (be it digitally or not). In some cases, this classification is broken down in the level of geometrical objects (or the level of model detail) and the level of Information Objects (or the Level of Information Detail). The concept of Building Lifecycle Stages (BLS) has existed for numerous decades and is used in standards and regulations, such as RIBA, HOAI, etc. This concept aims to break down the activities required to design, construct, commission, and operate a building and to allocate sets of activities to well-defined roles or stakeholders involved in the BLS.

Deliverable D3.3 of the BIM4EEB project aims to extend the above mentioned open meta-data building model of the AECO-domain with features aiming to support the management of LOD over the various stages of the BLS corresponding to renovation process modelling. This shall be achieved through the development of an ontology framework that is part of a larger modular ontology platform developed in the BIM4EEB project.

The discussion starts with a specification of this deliverable in the context of the BIM4EEB project, i.e. oblective, what are the relations to other tasks and deliverables in the BIM4EEB project. This is followed by a state-of-the-art analysis (SOTA) in the area of LOD. Based on this SOTA the authors present an ontological approach for the representation of different LOD systems. In a subsequent chapter, the authors will then discuss requirements for LOD-based BIM-data representation, i.e. we clearly distinguish between the “building objects” which usually exist over multiple BLS and data specifying the “building objects” which might be distinct over the different BLS and thus evolve over those LODs. Based on the analysis, the ontology approach is further extended to address these requirements and enable to represent information in multiple levels of detail.

Furthermore, a representational schema comprised of various components in renovation workflow is developed. These components are LOD, BLS, BIM data, renovation activities, stakeholders and use cases. Along with the renovation component description, the representational schema also defines the relationship between these components corresponding to the renovation workflow. The ontology approach is further extended to represent these components and their interrelations. The final output of this deliverable is a Digital Construction Lifecycle (DICL) ontology to represent BIM data in different levels of detail corresponding to renovation workflow. Finally, D3.3 briefly discusses aspects of developed ontology management and also presents a simple demonstration example.

PUBLISHING SUMMARY

Deliverable D3.3 focuses on BIM data representation in different Levels of Detail (LOD) corresponding to the renovation process modelling. The modelling approach in this deliverable corresponds to the renovation workflow processes specified in Task T 2.1 of BIM4EEB. It aims to address the current deficit in specifying the varying scope and detail of product models over the different lifecycle stages. This difference of scope and detail results from the various information requirements of the distinct stakeholders being actively involved in the several building lifecycle phases.

The development goal is a Digital Construction Lifecycle (DICL) ontology which describes the different activities, tasks, and sequences and their interrelations to stakeholders involved and the information required. The DICL ontology is linked to other BIM4EEB ontologies (D3.2 and D3.4) to support the representation of BIM data in different Levels of Detail.

The DICL ontology proposed will become a part of the BIM4EEB modular ontology framework (D3.6). It establishes an effective and transparent instrument for the classification and analysis of product and process data related to LOD and BLS corresponding to roles and stakeholders (designers, architects, vendors, contractors, workers, occupants), including their capabilities and responsibilities to manage and maintain LOD-data. Additionally, the instruments provided will support the establishment of standardised collaboration protocols between stakeholders, uninterrupted communication flows and sharing of relevant data and information represented in BIM models when required.

The proposed approach will allow for modular, incrementally adaptable, efficient, customised data-sharing features and thus the provision of high-value applications and tools, not only towards optimising design and decision-making but also for increasing the effectiveness of on-site works. This task will deliver an ontology for BIM data representation on different levels of development corresponding to renovation process modelling (D3.3).

Table of Contents

1 Introduction and Background 11
1.1 BIM4EEB Context 12
1.2 Structure of the Document 14
1.3 Components for analysis 14
1.4 DICL ontology introduction 16
1.5 Major updates compared to previous version 18
2 State-of-art analysis on LOD 19
2.1 LOD Definition in European Standards 19
2.2 LOD Definitions in International Standards 21
2.3 Flexible Usage and Adaptability 22
2.4 Requirements for ontologies to support LOD systems 22
3 Ontology-based LOD representation 24
3.1 LOD – Ontology representation 24
3.2 LOD - Demonstration 25
3.3 LOD - Verification 27
4 LOD-based BIM data Management 28
4.1 Approach Analysis 28
4.2 Ontological representation 30
5 Renovation Process Modelling 33
5.1 Renovation components 33
5.2 Representational Schema 35
5.3 Ontological representation of renovation workflow 36
5.4 Building Lifecycle Stages (BLS) 39
6 Ontology Management & Validation 46
6.1 External ontology alignment components 46
6.2 Dependency on BIM4EEB ontologies 47
6.3 Validation of the ontology 47
7 Summary and Conclusion 48
8 Publication bibliography 49
9 Annex I – LOD framework validation 53
10 Annex II – BLS framework validation 58
11 Annex III – Integrated Product-Process modelling examples 59
12 Annex IV – Additional Use Case Definitions 62
13 Annex V - Basic demonstration of LOD sensitive BIM data 63

List of Abbreviations

AECO	Architecture, Engineering, Construction and Operation
AIA	American Institute of Architects
BIM	Building Information Model(ling)
BIM4EEB	BIM-based fast toolkit for Efficient rEnovation in Buildings
BLS	Building Lifecycle Stages
DICL	Digital Construction Lifecycle
cf.	An abbreviation for the Latin word “confer”, used to refer to other material or ideas which may provide similar information or arguments
e.g.	Latin abbreviation stands for "exempli gratia" and means “for example”
EU	European Union
HVAC	Heating, Ventilation, Air Conditioning
i.e.	Latin abbreviation for "id est" and means “in other words”
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
LOD	Level of Model Detail / Level of Development
LoD	Level of Definition
LOI	Level of model Information / Level of Information
LOG	Level of Geometry
ME	Model Elements
MS	Microsoft ®
MVD	Model View Definitions
RDF	Resource Description Framework
RIBA	Royal Institute of British Architects
SPARQL	Protocol and RDF Query Language
UML	Unified Modelling Language
w.r.t.	Abbreviation for "with respect to"
WP	Work Package
CQ	Competency Question

List of Figures

Figure 1: Components of Renovation (Product-Process) Modelling 15
Figure 2: DICL Ontology – Class Taxonomy 17
Figure 3: DICL Ontology – Data Property Taxonomy 17
Figure 4: Ontology-based LOD framework 24
Figure 5: Relation between the LOD Classification-Levels 26
Figure 6: Relationship between the LOD scales 26
Figure 7: Different levels of property representation 29
Figure 8: Building Product Data management 31
Figure 9: Agent's involvement in activities w.r.t BLS 33
Figure 10: Representational schema - Renovation Product-Process representation 36
Figure 11: Renovation Workflow – Ontology representation 37
Figure 12: Information flow roles within the activities 38
Figure 13: Dynamic allocation of Activities to Use Cases 39
Figure 14: BLS in ontology representation 41
Figure 15: Demo1 – Relation between main stages of (BS EN 16310:2013) BLS framework 42
Figure 16: Demo2 – Relation between sub-stages of (BS EN 16310 2013) BLS framework 43
Figure 17: Dependency Schema for BIM4EEB ontologies 47
Figure 18: LOD levels according to the USA BIMForum System 56
Figure 19: Relation of LOD Classification-Levels with other levels of the USA LOD scale 56
Figure 20: Next levels for each level in the USA LOD system 57
Figure 21: Previous levels for each level in the USA LOD system 57
Figure 22: The next stages of the stage "Construction" according to BS EN 16310 58
Figure 23: Previous stages of the stage "Construction" according to BS EN 16310 58
Figure 24: BIM entities – LOD matrix example 59
Figure 25: Stakeholders and their Activities w.r.t. BLS 60
Figure 26: Stakeholders and their Activities w.r.t. information use 61
Figure 27: Demo BIM Model 63

List of Tables

Table 1: Defined BLS according to BIM4EEB Project (D2.1) ... 12
Table 2: List and number of the Stakeholders identified ... 13
Table 3: Ontology namespace parameters ... 16
Table 4: Other BIM4EEB ontologies represented in this deliverable ... 16
Table 5: LOD system according to the different national specifications ... 20
Table 6: Specific definition of LOD Levels as per BIM-Forum ... 21
Table 7: LOD - Queries and their results – Part 1 ... 27
Table 8: Stakeholders and their involvement in Use Case 1 ... 34
Table 9: BLS in different standards and publications ... 40
Table 10: BLS - Queries and their results – Part 1 ... 45
Table 11: External ontologies aligned with DICL ontology ... 46
Table 12: Alignment components with external ontology classes ... 46
Table 13: Alignment components with BIM4EEB ontology classes ... 47
Table 14: LOD - Queries and their results – Part 2 ... 53
Table 15: Stakeholders and their involvement in Use Case 2 ... 62
Table 16: Stakeholders and their involvement in Use Case 3 ... 62
Table 17: Comparison of BIM Model Instances ... 63
Table 18: Comparison of BIM Model attributes ... 65
Table 19: The query for the extraction of wall object data ... 66
Table 20: The query for the construction of wall object data ... 67
Table 21: Example 1 - Instances for the demonstration ... 69
Table 22: Object property extraction ... 71
Table 23: Construction details Extraction ... 73
Table 24: Example 2 - Instances for the demonstration ... 74

1 Introduction and Background

Architecture, Engineering, Construction and Operation (AECO) industry is a collaborative environment with the involvement of multiple disciplines and activities throughout the building lifecycle process (Karlapudi et al. 2020). This collaboration requires the iterative and cooperated exchange of information, and improves the building design over multiple lifecycle stages (Abualdenien and Borrmann 2018; Abualdenien and Borrmann 2019).

The collaboration and sharing of product information throughout the project lifecycle stages and among the project partners are fundamental to ensure the successful implementation of a project (Mangialardi et al. 2017; Saaksvuori and Immonen 2004). This management of the project's lifecycle information also ensures the reduction of error-prone operations, data communication problems (Mangialardi et al. 2017), and provides significant efficiency benefits, time-saving, etc (Di Biccari et al. 2018).

Since the last decade, Building Information Modelling (BIM) is an emerging approach and an enhanced business process in the AECO Industry (Allan and Menzel 2009; Li et al. 2017). This technical advancement aimed to improve the collaboration and data sharing between the stakeholders involved in construction projects (Keller et al. 2008; Zadeh et al. 2017).

The developments of the OpenBIM IFC meta-model (ISO 16739-1:2018; Venugopal et al. 2012) provide a building block for flexible interoperability of the BIM data. However, results from research (Karlapudi and Shetty 2019; Karlapudi and Menzel 2020; Dong et al. 2007; Guzmán and Zhu 2014; Lilis et al. 2017; Ramaji and Memari 2016) on IFC-based interoperability comprehensively explain the reasons for inadequate interoperability concerning BIM model development, the difference of domain-based knowledge representations and complexity in the extraction of data from Model View Definitions, etc.

Concepts like Level of Development (LOD), Information Delivery Manual (IDM), Model View Definitions (MVDs), and others were developed as part of the BIM process to address and support the diverse use-cases and their requirements needed in a construction project (Treldal et al. 2016; Xu et al. 2020). The defined LOD levels from the different national standards or code of practices aim to describe the granularity and the sequential refinement of both geometric and semantic information about an object (Hooper 2015). This methodological approach assists to track the improved changes or refinements of the building objects throughout the different phases of the building design and enables to set the required level of data export or import between the multiple disciplines.

Despite the improvement, there is a lack of successful implementation and management of LOD functionalities within existing BIM solutions (Papadonikolaki et al. 2018). One main issue is the insufficient understanding of diverse frameworks for the adoption and representation of LOD levels. Based on the analyzed knowledge, this document provides a flexible approach to define information levels according to different available standards and aligned to the use-case requirements (IDM concept) involved in construction projects. This approach is majorly based on the linked data and ontology concepts.

Furthermore, the structured representation of building data in ontologies (triples or graphs) enables stakeholders to semantically interpret data for various domain-specific operations with minimal human interventions. This framework is also easily adaptable and applicable to the present BIM process (Beetz et al. 2009; Pauwels et al. 2017; Chen et al. 2018).

1.1 BIM4EEB Context

Building Information Modelling (BIM) is an advanced information technology that should be adopted in the AEC industry by relevant stakeholders. These stakeholders are performing their activities with the use of BIM models (Karlapudi et al. 2020). Throughout Building Lifecycle Stages (BLS) a renovation project has several phases, e.g. early design, initial design, design of documentation, detailed design, before procurement etc. Going through the BLS the BIM model evolves. Early BIM models are broader and focus on interfaces between building and spatial 'objects'. In subsequent steps, the refined models focus on smaller parts but specify those parts and their internal structure in increasing detail and with enriched information content. This information refinement process is specified through the concept of "Levels of Details" (LOD).

In the context of the BIM4EEB project, this deliverable is dedicated to developing an ontology to represent these data refinements in different LOD corresponding to the renovation process modelling. This ontology also represents the renovation process by describing the involved activities, activity sequences, their interrelations to stakeholders and the required information. For the purpose of ontology definition, some of the defined concepts from the other BIM4EEB ontologies (i.e. from D3.2, D3.4) are used.

1.1.1 References to previous deliverables

Building Lifecycle stages: the initial analysis was completed within the previous deliverables of work packages 2, but Section 5.4 of this report further progressed with the analysis of more standards and summarises the findings. The table below indicates the identified stages within the Deliverable D2.1.

Table 1: Defined BLS according to BIM4EEB Project (D2.1)

BLS No	BLS Category/sub-category Name
0	Initiative
0.1	Market study
0.2	Business case
1	Initiation
1.1	Project initiation
1.2	Feasibility study
1.3	Project definition
2	Design
2.1	Conceptual Design
2.2	Preliminary Design
2.3	Developed Design (B&I)
2.4	Technical Design
2.5	Detailed Design
3	Procurement
3.1	Procurement
3.2	Construction Contracting
4	Construction
4.1	Pre-construction
4.2	Construction

4.3	Commissioning
4.4	Hand Over
4.5	Regulatory Approval
5	Use
5.1	Operation
5.2	Maintanence
6	End of Life
6.1	Revamping
6.2	Dismantiling

Relevant activities and involved stakeholder: From the analysis of well-defined 196 activities explained in the deliverable D2.1, chapters 3 and 4, Tables 4-26, we have structured the principal stakeholders as presented in the following Table 2. It is necessary to admit that only stakeholders 1 – 22 were mentioned in the D2.1 renovation process workflow. Furthermore, stakeholders 31 – 38 were mentioned only as “involved in the renovation process” (see BIM4EEB D2.1 section 2.2), but not reviewed in the workflow chart (D2.1, chapter 5). However, to maintain the scope and keep the methodology of this D3.3 report open, this list of stakeholders will be used for the subsequent developments within this deliverable.

Table 2: List and number of the Stakeholders identified

№	Stakeholders	№	Stakeholders	№	Stakeholders
1	Inhabitant / End-user	14	Building Services Designer	27	Interior Designer
2	Client / Owner	15	Sustainability Adviser	28	Lighting Designer
3	Client' Adviser	16	Facilities Mgmt. Advisor	29	Security Adviser
4	Project Leader	17	Technical Adviser	30	Access Consultant
5	Site Surveyor	18	Contractor	31	Supplier
6	Cost Consultant / QS	19	Local Authority	32	Master Planner
7	Health and Safety Adviser	20	Bank or third-party financier	33	Planning Consultant
8	Information Manager	21	Work supervisor	34	Party wall surveyor
9	Contract Administrator	22	Tester (Commissioner)	35	Sub-Contractor
10	Tenderer	23	Fire safety designer	36	Maintenance Planner
11	Lead Designer	24	Landscape designer	37	Construction Leader
12	Architectural Designer	25	Acoustic Consultant	38	Operational Leader
13	Structural Designer	26	Cladding Specialist		

1.1.2 Related publications

A set of scientific papers were published in the peer-reviewed conferences based on the achieved results from this deliverable. These scientific publications are listed below.

1. KARLAPUDI J., Menzel K., Törmä S., Hryshchenko A., Valluru P. (2020) **Enhancement of BIM Data Representation in Product-Process Modelling for Building Renovation**. In: Nyffenegger F., Ríos J., Rivest L., Bouras A. (eds) *Product Lifecycle Management Enabling Smart X. PLM 2020. IFIP Advances in Information and Communication Technology*, vol 594. Springer, Cham.
2. Karlapudi, J., Valluru. P., Menzel. K. (2021) **Ontology approach for Building Lifecycle data management**. *In: Proceedings of the 2021 ASCE International Conference on Computing in*

Civil Engineering (i3CE2021) - IT for Smart Infrastructure and Communities, 12 - 14 September 2021, Orlando, Florida USA.
3. Karlapudi, J., Valluru. P., Menzel. K. (2021) **Ontological approach for LOD-based BIM-data management**. *In: Proceedings of the LDAC2021 - 9th Linked Data in Architecture and Construction Workshop,* 11 - 13 October 2021, Luxembourg.

1.2 Structure of the Document

The deliverable starts with an introduction that describes the scope and context of the deliverable along with the relation to other deliverables or concepts of the BIM4EEB project. From the task description, this deliverable aimed to develop an ontology to represent the BIM data in different LODs corresponding to renovation process modelling.

To achieve this deliverable object, it is important to understand the structure and framework of LOD systems. So the chapter 2 of this deliverable focused on the state-of-the-art analysis of LOD systems or standards. This comprehensive understanding further helps to develop the ontology representation of LOD systems which is the clear focus of chapter 3. Along with the ontology-based representation of LOD systems, chapter 3 also provides a clear explanation of the developed ontology representation through the example demonstration and verification.

Another important task in the deliverable is to link this LOD system to the BIM data represented through ontologies. For this purpose, an investigation on different possibilities and requirements for LOD-based BIM data representation is carried. Based on the analysis and project requirements a flexible methodology called “objectification of properties” is considered to represent the BIM data in different levels of detail. Chapter 4 illustrates this development along with clear explanations.

The further development of this deliverable is the renovation process ontology which aimed to represent the renovation activities, their sequences and the involved stakeholders. This renovation process ontology also should be linked to the ontologies representing the information requirements in different LODs. To achieve this goal, in Chapter 5, an analysis is conducted on the defined renovation workflow in Deliverable D2.1. Based on this analysis results, an ontology representation is developed to indicate the renovation processes and link them to the BIM data represented in different levels of detail. The emphasis on renovation process modelling provides a potential link to subsequent work in Task 3.4 of WP3, BIM4EEB.

With these developments, the major tasks are achieved and successfully document in this deliverable. In addition to the developments, chapter 6 provides information on the management of the developed ontologies. Finally, chapter 7 draws conclusions on the developed and demonstrated work.

1.3 Components for analysis

The components listed below specify the scope of our work and introduce major concepts for the modelling activities in Task 3.3. These components further guides the work in T3.3 and supports the development of the Digital Construction Lifecycle Ontology (see also Figure 1).

- **LOD** for required BIM data. This is a modelling approach representing the evolution of data in BIM.
- **Stakeholders** acting in renovation processes – ∑38 were selected and explained in D2.1 (see also Table 2). In this report, we distinguish stakeholders as:
    - Information Provider: These are stakeholders collecting, compiling, creating information/data to be further used by Information Processor(s).
    - Information Processor: These are stakeholders receiving, retrieving, changing, analysing the information before transferring it to other stakeholders acting as Information Consumers.
    - Information Consumer: These are stakeholders receiving and consuming the information.
- **BIM-**objects. These are modelled in IFC as classes with corresponding properties or sets of properties. BIM-objects are used to model building products and processes.

- **Building Lifecycle Stages** (BLS): This is a modelling approach to break down the processes to design, construct, operate, or demolish a building. BLS are closely related to rules and regulations.
- **Activities:** This is a modelling approach to represent dedicated actions and groups of them (processes) which are executed by distinct stakeholders to design, construct, operate, or demolish a building. 196 activities related to building renovation are outlined by D2.1.
- **Use Cases:** This is an approach supporting the dynamic combination of Activities. Three Use Cases are proposed in this deliverable.

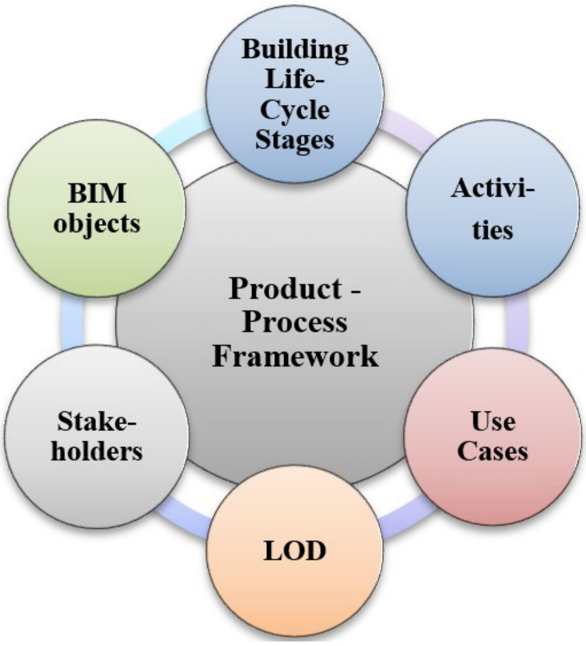


Figure 1: Components of Renovation (Product-Process) Modelling

The above Figure 1 is reflecting those six components/domains to be considered for the definition of the "Digital Construction Lifecycle (DICL)" ontology. The layout represented in Figure 1 is defining a 6-dimensional space where all of those components are related to each other.

Along with the above components, the following standards and regulations are referred to in this report because some or all of their content complies with the requirements of this deliverable. For dated references the latest edition of the referenced document (including any amendments) applies:

- EN ISO 29481-1: Building information models - Information delivery manual - Part 1: Methodology and format, (ISO 29481-1:2016);
- ISO 23386:2019: Building information modelling and other digital processes used in construction — Methodology to describe, author and maintain properties in interconnected dictionaries (ISO 23386:2020);
- ISO 23387: Building Information Modelling (BIM) - Data templates for construction objects used in the lifecycle of any built asset - Concepts and principles (ISO 23387:2020);
- ISO 6707-1: Buildings and civil engineering works - Vocabulary - Part 1: General terms (ISO 6707-1:2017);
- BS EN 17412:2019: Building Information Modelling - Level of Information Need - Concepts and principles; German and English version (BS EN 17412-1:2020).

1.4 DICL ontology introduction

This section provides an initial introduction to the ontology developed as a part of task 3.3. The table below describes details on developed DICL ontology regarding the repository, namespace convention, etc.

Table 3: Ontology namespace parameters

Parameter	BIM4EEB considerations
Ontology name	Digital Construction Lifecycle (DICL) ontology
Prefix	dicl
Namespace	https://w3id.org/digitalconstruction/0.5/Lifecycle#
Repository	https://github.com/digitalconstruction/Lifecycle/
Html webpage	https://w3id.org/digitalconstruction/0.5/Lifecycle

The present version of the ontology is represented with the name "Digital Construction Lifecycle" ontology and uploaded to the Digital Construction GitHub platform. The HTML documentation of the ontology is developed with the help of the pyLODE 2.8.5 tool (Car 2021) and published under the Digital Construction Ontologies webpage and licensed by Creative Commons 4.0 international license.

Table 4: Other BIM4EEB ontologies represented in this deliverable

Ontology	Prefix	Namespace
Contexts	dicc	https://w3id.org/digitalconstruction/0.5/Contexts#
Variables	dicv	https://w3id.org/digitalconstruction/0.5/Variables#
Entities	dice	https://w3id.org/digitalconstruction/0.5/Entities#
Process	dicp	https://w3id.org/digitalconstruction/0.5/Processes#
Agents	dica	https://w3id.org/digitalconstruction/0.5/Agents#
Information	dici	https://w3id.org/digitalconstruction/0.5/Information#
Lifecycle	dicl	https://w3id.org/digitalconstruction/0.5/Lifecycle#
Levels	diclvl	https://w3id.org/digitalconstruction/0.5/Levels#
Stages	dicstg	https://w3id.org/digitalconstruction/0.5/Stages#

Hereafter, any concept represented with the prefix "dicl" indicates their belongingness to the Digital Construction Lifecycle (DICL) ontology. Table 4 is dedicated to indicating the elaborated information regarding the prefixes of ontology classes.

Within this deliverable, the Ontology Visual Notations from (Garijo and Poveda-Villalón 2020) are adopted to represent the DICL ontology. This representation is performed in two different phases. In the introduction phase, only the taxonomy of ontology structure corresponding to Classes and Data properties is represented (see Figure 2, Figure 3 overleaf).

Later in the development phase, the representation is elaborated regarding the relations between the classes using object properties based on the investigated requirements. In the development process of DICL ontology, the classes from the other BIM4EEB workflow ontologies are also used. The major motivation is to reduce the redundancy and repetition of ontology concepts within the BIM4EEB onology suite.

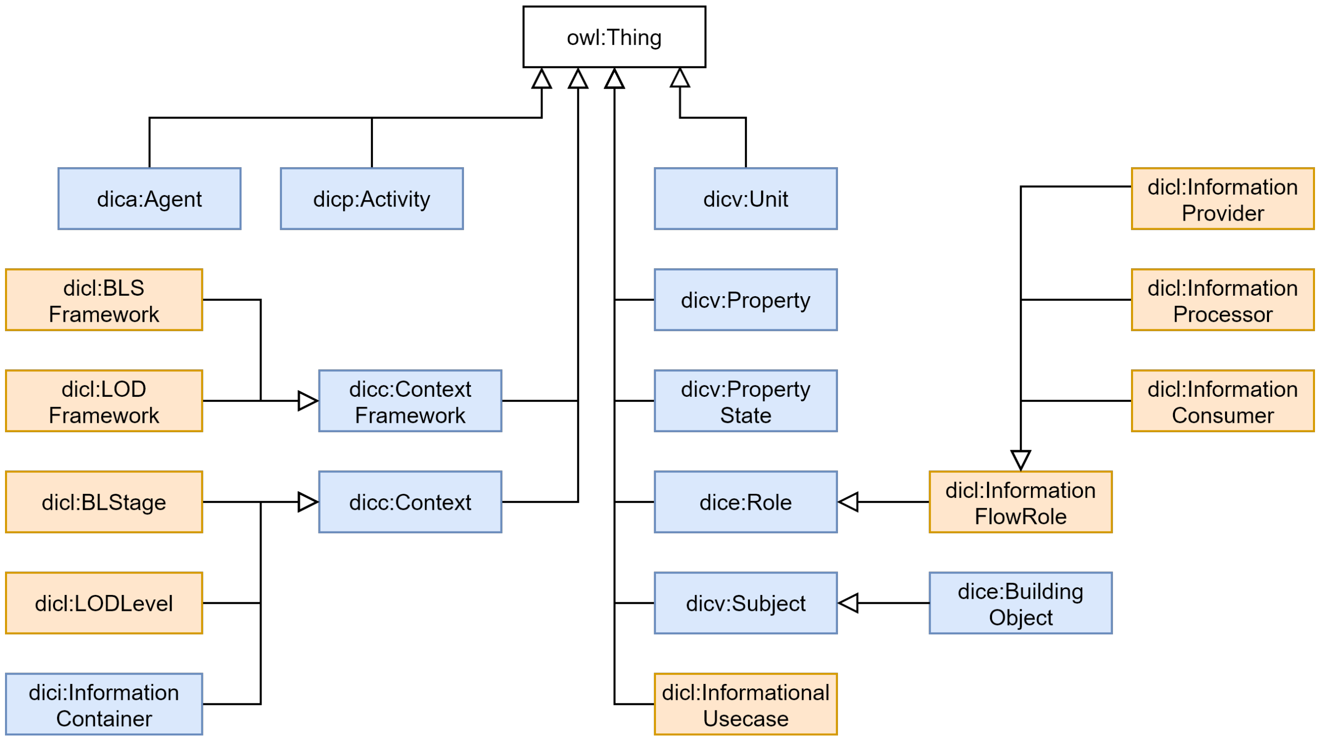


Figure 2: DICL Ontology – Class Taxonomy

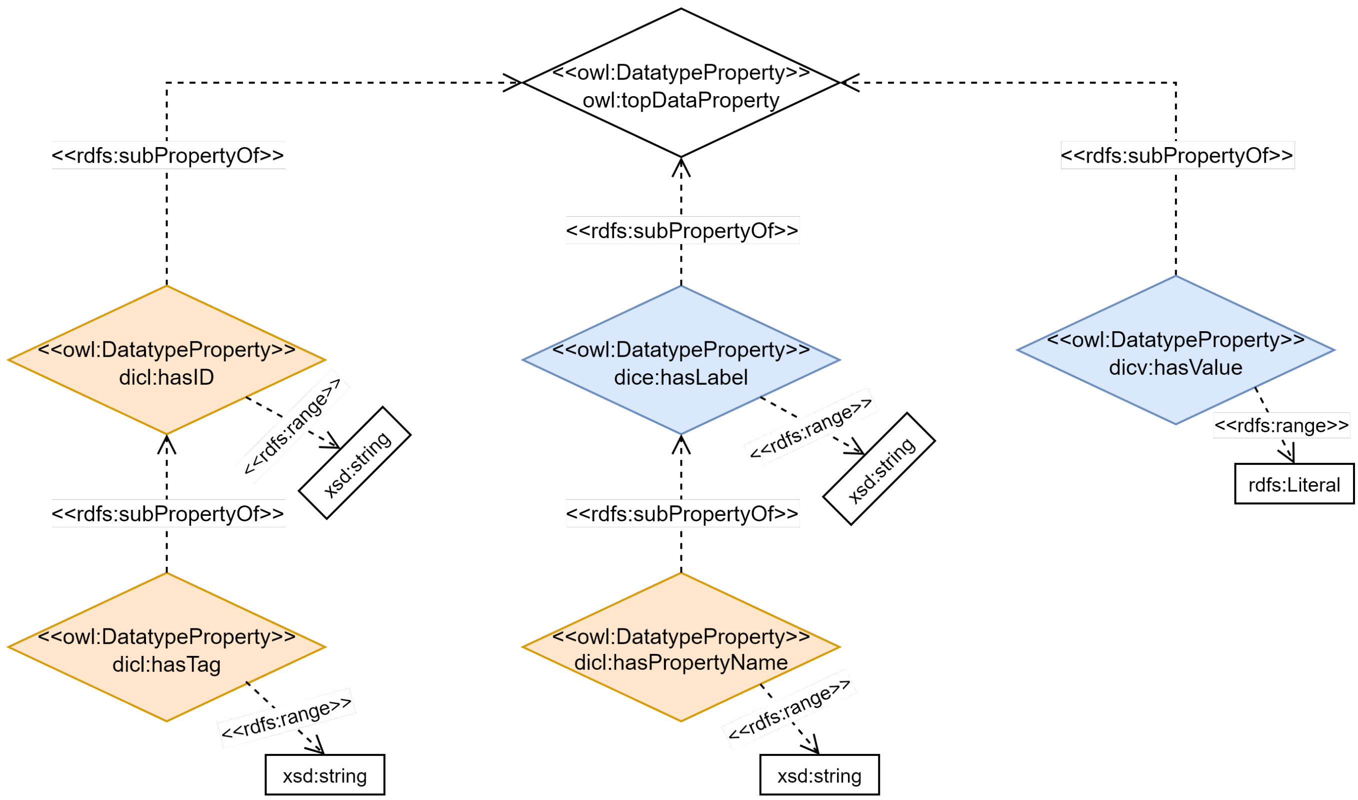


Figure 3: DICL Ontology – Data Property Taxonomy

1.5 Major updates compared to previous version

A list of major updates addressed in this deliverable when compared to the previous version (version 2.2) is presented below.

- Extended the discussion related to competency questions
- Readjusting the subheadings for chapters 3, 4 and 5
- Rearranging the subsections of Chapter 5
- Adjusted all figures to chowlk notations
- Added a conclusory section (4.1.3) to subsection 4.1

2 State-of-art analysis on LOD

The abbreviation “LOD” is used in various meanings. When writing the proposal, we were strongly influenced by the LOD-definition of the American Institute of Architects (AIA), which is commonly known as “*The LOD framework for the Building Information Modelling Protocol Form*“ (AIA Contract Documents 2013). In this case, the term LOD refers to the “Level of Development” required for BIM-model elements’ content. The term “Level of Development” is used rather than “Level of Detail” in recognition of the fact that a visually very detailed element might in fact be generic and despite appearances might be at a low level of design development (BIM Wiki 2020).

The LOD levels of BIM data should be generally defined for different stages of projects when data sharing takes place. This is a pragmatic approach to refer to the granularity of BIM data and to track the data refinements throughout the project progression over multiple stages of the building. Furthermore, this would allow stakeholders to verify that project information is detailed enough to meet their requirements, enabling them to decide whether to proceed to the next project stage or not (Karlapudi et al. 2020).

Commencing in 2004, different countries have developed different LOD standards generating a complex situation at an international level. A single, unified approach for the definition of LOD is pending. The abbreviation “LOD” is used in various meanings in different countries, such as the USA - BIMForum Specification (BIMFORUM), UK - BS 1192-1 and PAS 1192-2, 3 (PAS 1192-2) (PAS 1192-3), and Italy UNI 11337 part 4 (UNI 11337-4) (see also Table 5).

An analysis of various LOD systems according to the standard of practices or national norms is carried out as a part of the initial study. Based on the results from this analysis a conclusion is drawn on flexible usage of these frameworks into the construction project irrespective of standards and the region of work. It also identified the basic requirement that needs to be addressed in the ontology representation of the LOD frameworks or systems.

2.1 LOD Definition in European Standards

An initial analysis of LOD had been carried out in Task 3.1 (see Deliverables D3.1 section “*3.4 Evolution of information during execution*”).

In this deliverable, we provide a comparison between selected LOD scales referred to in different stages of the project. More specifically, we illustrate the different LOD framework definitions from the following standards based on national or regional specifications:

– Italian - UNI 11337 in its part 4 (UNI 11337-4),
– EN 17412:2019: “BIM - Level of Information Need - Concepts and principles
– UK - PAS 1192-2 (PAS 1192-2:2013) and 3 (PAS 1192-3:2014),
– EN 19650 (BS EN ISO 19650-1, 2018)

A relevant standard under development from WG2 of CEN/TC 442 is EN 17412:2019: “Building Information Modelling - Level of Information Need - Concepts and principles” (BS EN 17412-1:2020). This document specifies the methodology to describe the Levels of Definitions of BIM deliveries throughout the lifecycle of built assets, and in particular how the Level of Information is related to the different BIM usages. It clarifies how far a common European definition can be agreed upon and when further detailing must occur within each member state. (BS EN 17412-1:2020) defines two terms used:

- LOG: Level of Geometry, and LOI: Level of Information.

This is to describe the state of the definition of Model Elements (ME) in Building Information Models. It establishes a common way to label the different levels, here, in particular, the Levels of Geometry and Levels of Information, according to a naming scheme, such as numeric as 1-5, or 100-500, or alphanumeric, such as A-E, or by a combination of both.

The recently published standard EN 19650 (BS EN ISO 19650-1, 2018), which was developed on the basis of the British national standard BS 1192-2 “Specification for information management for the capital/delivery phase of construction projects using building information modelling” (PAS 1192-2, 2013) introduces LoD as “Level of Definition”.

To make the case even more confusing, it provides the following “formula”: LoD = LOD + LOI.

The meaning of this “formula” is, that the Level of Definition comprises of two components, such:

- Levels of model Detail (LOD), which relates to the graphical content of models;
- Levels of model Information (LOI), which relates to the non-graphical content of models.

The levels of model detail and model information are generally defined for key stages of a project at which “data drops” (information exchanges) take place, allowing stakeholders to verify that project information is consistent with their requirements and enabling them to decide whether to proceed to the next stage.

However, at present, there is no standardised definition for the timing of data drops or for levels of model detail and model information, other than the suggestion that they should be aligned to stakeholders’ decision points and should be consistent across all appointments. This is because it is thought they will vary depending on the nature of the project. However, some very broad guidance on how to define data drops was given in (PAS 1192-2, 2013).

Table 5 provides an initial comparison between three available standards, such as the Italian - UNI 11337-4, EN 17412:2019 and the US-based definition of LOD, published and maintained by the BIM-Forum with major contributions from AIA and the Associated General Contractors of America (AGC).

EN 19650 was not explicitly included in the table, since it uses definitions that are comparable to those provided in EN 17412:2019.

Table 5: LOD system according to the different national specifications

Region Standard	LOD means	Classification-Levels	Levels or Scales
Italy: UNI 11337-1: (2017)	Level of Development of Objects	LOG: Geometrical Objects	LOG A, LOG B, LOG C, LOG D, LOG E, LOG F, LOG G
		LOI: Information Objects	LOI A, LOI B, LOI C, LOI D, LOI E, LOI F, LOI G
Europe: EN 17412: (2019)	Level of Definition	LOD: Level of model detail	LOD 1, LOD 2, LOD 3, LOD 4, LOD 5, LOD 6
		LOI: Level of Information detail	LOI 1, LOI 2, LOI 3 LOI 4, LOI 5, LOI 6
USA: LOD: BIM Forum (2020)	Level of Development	LOD: As Designed	LOD100, LOD200, LOD 300, LOD 350, LOD 400
		LOD: As Built	LOD 500

2.2 LOD Definitions in International Standards

The BIM-Forum provides the following background information on their webpage:

> *"The* Level of Development (LOD) Specification *is a reference that enables practitioners in the AEC Industry to specify and articulate with a high level of clarity the content and reliability of Building Information Models (BIMs) at various stages in the design and construction process.*
> *The* LOD Specification *utilizes the basic LOD definitions developed by the AIA for the* AIA G202-2013 Building Information Modeling Protocol Form*[1] and is organized by CSI Uniformat 2010[2].*
> *It defines and illustrates characteristics of model elements of different building systems at different Levels of Development. This clear articulation allows model authors to define what their models can be relied on for, and allows downstream users to clearly understand the usability and the limitations of models they are receiving."* It also says "*It does not prescribe what Levels of Development are to be reached at what point in a project but leaves the specification of the model progression to the user of this document."*

Naming conventions and LOD specifications as per (AIA Contract Documents 2013) and (BIM FORUM 2020) are compiled and presented in Table 6 (below). One clear benefit of these definitions is an evaluation of the scope and quality of information; e.g. by saying "*Any information derived from LOD 100 elements must be considered approximate.*"

Table 6: Specific definition of LOD Levels as per BIM-Forum

LOD Level	LOD Description
LOD 100	Symbolic and not geometric representation. Examples are information attached to other model elements or symbols showing the existence of a component but not its shape, size, or precise location. Any information derived from LOD 100 elements must be considered approximate.
LOD 200	Generic system object or assembly – graphic representation (non-graphic information may be attached) with approximate quantities, size, shape, location and orientation. They may be recognizable as the components they represent, or they may be volumes for space reservation. Any information derived from LOD 200 elements must be considered approximate.
LOD 300	LOD 200 + System or Components - graphic representation (non-graphic information may be attached) with specific quantities, size, shape, location, orientation can be measured directly from the model without referring to non-modelled information such as notes or dimension callouts.
LOD 350	LOD 300 + Interfaces with other building systems added. Parts necessary for coordination of the element with nearby or attached elements are modelled. These parts will include such items as supports and connections.
LOD 400	LOD 350 + Information added – for detailing, fabrication, assembly and installation. A LOD 400 element is modelled at sufficient detail and accuracy for the fabrication of the represented component.
LOD 500	Field-verified representation in terms of size, shape, location, quantity, and orientation. It is not an indication of progression to a higher level of model element geometry or non-graphic information (which may also be attached to the Model Elements).

2.3 Flexible Usage and Adaptability

When using the LOD approach, it is necessary to highlight that this should be used as a methodological approach that can be adjusted and modified to different national constraints. The approach we take is defining the reasonable number of different levels of detail to specify what should be included for our purposes (e.g. building renovation), and amalgamating the information content (concerning geometrical and alpha-numerical data) in one commonly specified, holistic definition. This methodological approach is valid, irrespectively in what country a certain standardised solution applies. This allows us to provide a ‘robust framework’ as a solid foundation for LOD-ontology development.

Furthermore, the clarification should be made that LOD's relate to Model Elements and not complete Building Information Models. Furthermore, on the attribute level, one must be sure that the name of the attribute like “height” or “length” is always correctly interpreted in the ontology. This is usually achieved through alignments (see Deliverable D3.6).

To maintain the general applicability, it should always be possible to change the LOD scale in the LOD ontological structure with additional mapping represented by the abovementioned (BS EN 17412-1:2020). As an outcome, it will be possible to outline a subset of parameters in form of LOD-related matrixes that will satisfy data exchange requirements during selected stakeholders’ activities.

With this knowledge on different LOD standards, scales and their relations, we move to further investigation of the necessary requirements for the ontological representation of a LOD structure. The following sub-section aims to describe these requirements along with the possible options to develop a common ontological representation of LOD systems.

2.4 Requirements for ontologies to support LOD systems

In the area of information management, the support for LODs centres around the challenge to represent:

(1) various LOD systems,

(2) multiple versions of information about the same object and

(3) the connections of LOD-specific data to building lifecycle processes.

A proper representation not only allows the users to access and work inside one specific LOD but enable various cross-LOD functions:

- to access the history of values,
- to utilize the links, annotations, and other enrichments of previous LOD objects with those in subsequent LODs,
- to check the consistency and possible deviations between objects at different LODs, and
- to determine what kinds of adjustments to previous LOD models would be needed.

Moreover, it helps to connect LODs to other aspects of building information: to keep track of the origin of information and to maintain the rules for validating it against the requirements of subsequent activities. This analysis of LOD requirements suggests at least the following areas where ontology definitions are needed to properly support LODs:

- LOD frameworks: The representation of various levels, their relations, and the links to associated definitions.
- LOD sensitive BIM data: The representation of (versioned) properties of objects to capture the data at multiple different levels in an organized manner.
- Connection of the LOD framework to processes.

Competency questions.

Based on the requirement analysis a set of competency questions are developed and adopted in the ontology development process. In general, these competenc questions are addressing the functional requirements of the ontology. List of these competenc questions are presented below.

CQ1: How can the BIM data representation be adjusted or modified to different LOD systems?
CQ2: What is the link between the LOD system and its levels?
CQ3: What is the relation between the LOD Classification-Level and the LOD scale?
CQ4: What is the relation between LOD scales?
CQ5: How is the BIM object is represented?
CQ6: How to represent multiple versions of information about the same object?
CQ7: How the object properties and values for a specific LOD level are defined?
CQ8: What are the sources for LOD data?
CQ9: How the activities are defined in the renovation workflow?
CQ10: How to identify the sequence of activities?
CQ11: How are the stakeholders or agents related to activities?
CQ12: How does the renovation process linked to BIM data represented in LOD-sensitive manner?
CQ13: How to represent specific use cases within the renovation workflow?
CQ14: How to enable the representation of multiple BLS systems and/or stages?
CQ15: What is the link between the BLS system and its respective Stages?
CQ16: What is the relation between stages?
CQ17: What is the relation between the main-stage to sub-stage of other main-stage?
CQ18: How to represent the mapping between different BLS systems?

3 Ontology-based LOD representation

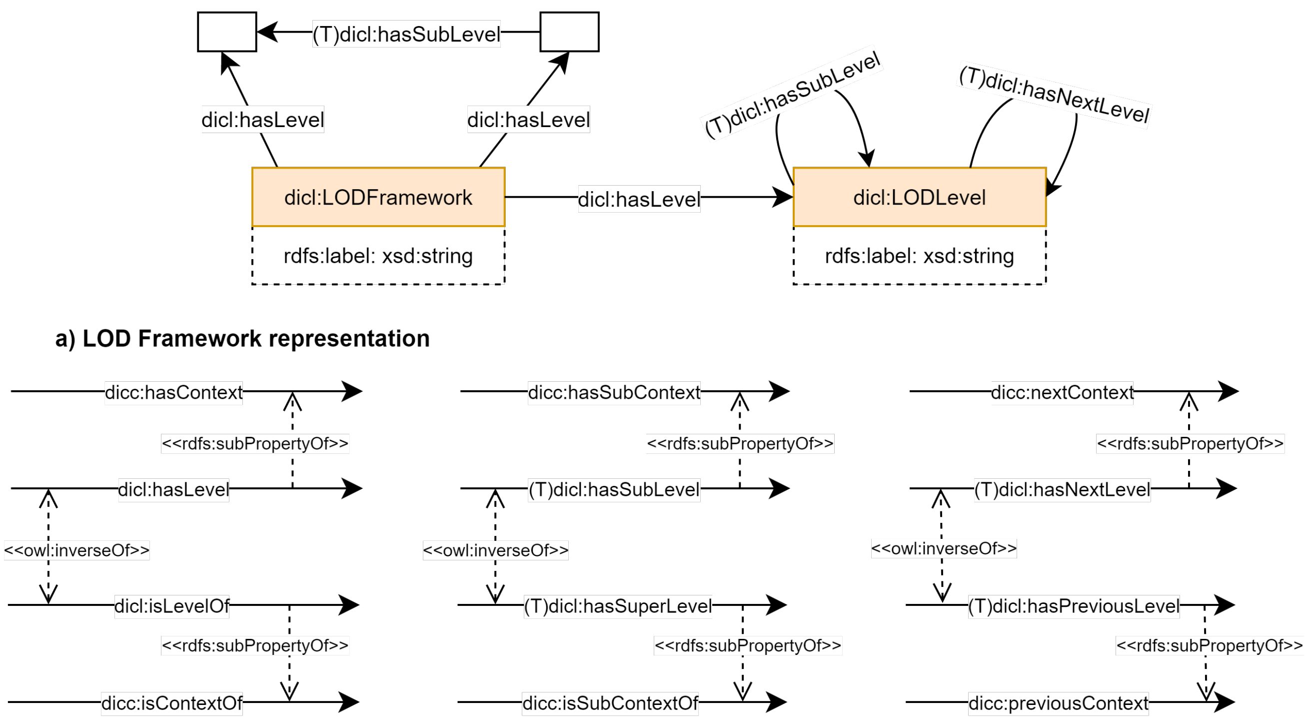
3.1 LOD – Ontology representation

The analysis of various LOD systems leads to different implementation requirements which need to be defined by the ontology-based frameworks. The concept of Competency Questions (CQ) is used to describe such requirements in terms of natural language questions. These CQs are used as the basis for the development of an ontology structure, concepts and properties. The development of an ontology framework for LOD representation is progressed towards providing all the answers for these competency questions.

- **CQ1:** How can the BIM data representation be adjusted or modified to different LOD systems?

From the general analysis of different LOD systems, an ontology schema is developed and illustrated in Figure 4. Since different renovation projects can adopt different LOD systems, the developed ontological structure of the LODs can accommodate the different standards of representations as detailed in Table 5.

The methodological approach is to represent LOD systems and their levels as classes, which can then be instantiated on a project-to-project basis. The class dicl:LODFramework can be instantiated with the frameworks (e.g. BIMForum, UK LOD, Italian LOD). The levels in the different frameworks are added as instances to the class dicl:LODLevel. These instances are subsequently assigned to BIM data and modified based on the project requirements.



a) LOD Framework representation

b) Object Property specification representation

Figure 4: Ontology-based LOD framework

- **CQ2:** What is the link between the LOD system and its levels?

As indicated in Table 5, each LOD system has several levels or scales which are classified majorly based on their purpose and the level of intent. For example, the UK and European systems classify their levels towards the representation of Graphical (Model) objects and Information objects. Instead, the USA system classifies their levels based on As Designed and As-Built information.

In the present approach, all these Classification-Levels and as well the scale of the LOD system are considered as Levels for the respective LOD system. To achieve this object, at first instance, the link between the LOD system and its respective Classification-Levels are generated using the object property dicl:hasLevel and its inverse property dicl:isLevelOf. Furthermore, a sub-property chain axiom (dicl:hasLevel o dicl:hasSubLevel → dicl:hasLevel) is assigned to the object property dicl:hasLevel to define semantic interpretation between the LOD scales and LOD Systems. This axiom inherits the new relation by saying LOD scales are defined as levels to the respective LOD system. Please refer to Figure 5 and Figure 6 for more understanding on defined and inherited relationships.

- **CQ3:** What is the relation between the LOD Classification-Level and the LOD scale? (Table 5).

Within the present approach, a sublevel relationship is generated between the LOD Classification-Level and its respective scale. The implementation idea is that the developed approach should fit into different classification criteria's available now or will be developed in future. This relationship is indicated in Figure 4 using the transitive object property dicl:hasSubLevel and its inverse property dicl:hasSuperLevel.

- **CQ4:** What is the relation between LOD scales?

According to the several LOD systems or frameworks, the level of growth in LOD levels is clearly based on the information growth or refinement. The same idea is considered in the generation of relationships between the levels of a LOD framework and the same is illustrated in Figure 4 using the transitive object property dicl:hasNextLevel and its inverse property dicl:hasPreviousLevel. A property characteristic called owl:TransitiveProperty is defined for these properties in order to represent the aggregation relationship between levels.

Apart from the development, an exemplary demonstration is presented in the below subsection to elaborate on the applicability and functionalities of the developed ontology framework.

3.2 LOD - Demonstration

For the demonstration, the BIMForum LOD framework is considered. The demonstration uses data from Table 5 and is represented according to the developed ontology structure which is illustrated in Figure 4.

As represented in Figure 5, the instances *diclvl:AsDesigned* and *diclvl:AsBuilt* are assigned as levels to the instance *diclvl:USA_BIMForum* using dicl:hasLevel object property. Since this object property's domain and range (dicl:hasLevel) are fixed to the classes dicl:LODFramework and dicl:LODLevel respectively, the inferencing engine automatically develops new knowledge by concluding the instance *diclvl:USA_BIMForum* belongs to the class dicl:LODFramework and the other instances are belonging to the class dicl:LODLevel. These defined and inherited relationships are clearly separated and illustrated in Figure 5 and Figure 6 by using seperate visualization diagrams.

Similarly, for other LOD systems, division of a level scale into different LOD Classification-Levels is also possible according to the developed framework. For instance, the UK LOD framework contains Level of Detail (LOD) for geometric information and Level of Information (LOI) for non-geometric information. They basically form two parallel sequences of levels.

Furthermore, the relationship between the neighbour instances *diclvl:AsDesigned* and *diclvl:AsBuilt* is assigned using dicl:hasNextLevel as a transitive object property. As defined in the ontology framework, the dicl:hasNextLevel object property has an inverse relationship assigned with dicl:hasPreviousLevel. This inherited knowledge (inverse relationship) is also represented in Figure 5.

The LOD scale information regarding each LOD Classification Level is defined by using the transverse object property called dicl:hasSubLevel and its inverse property dicl:hasSuperLevel. Because of the transverse nature of these object properties, any further sub-level of a LOD scale is also considered as a LOD level. Similarly, due to the defined axiom to dicl:hasLevel property, all these LOD scales are inferred as Levels (dicl:LODLevel) to the dicl:LODFramework (*diclvl:USA_BIMForum*).

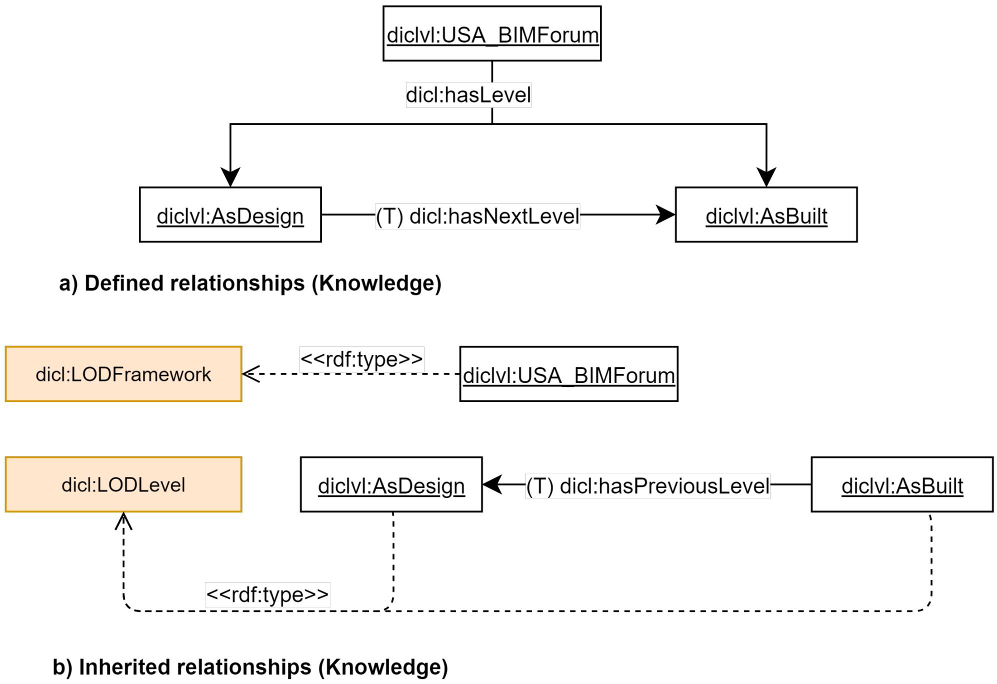


Figure 5: Relation between the LOD Classification-Levels

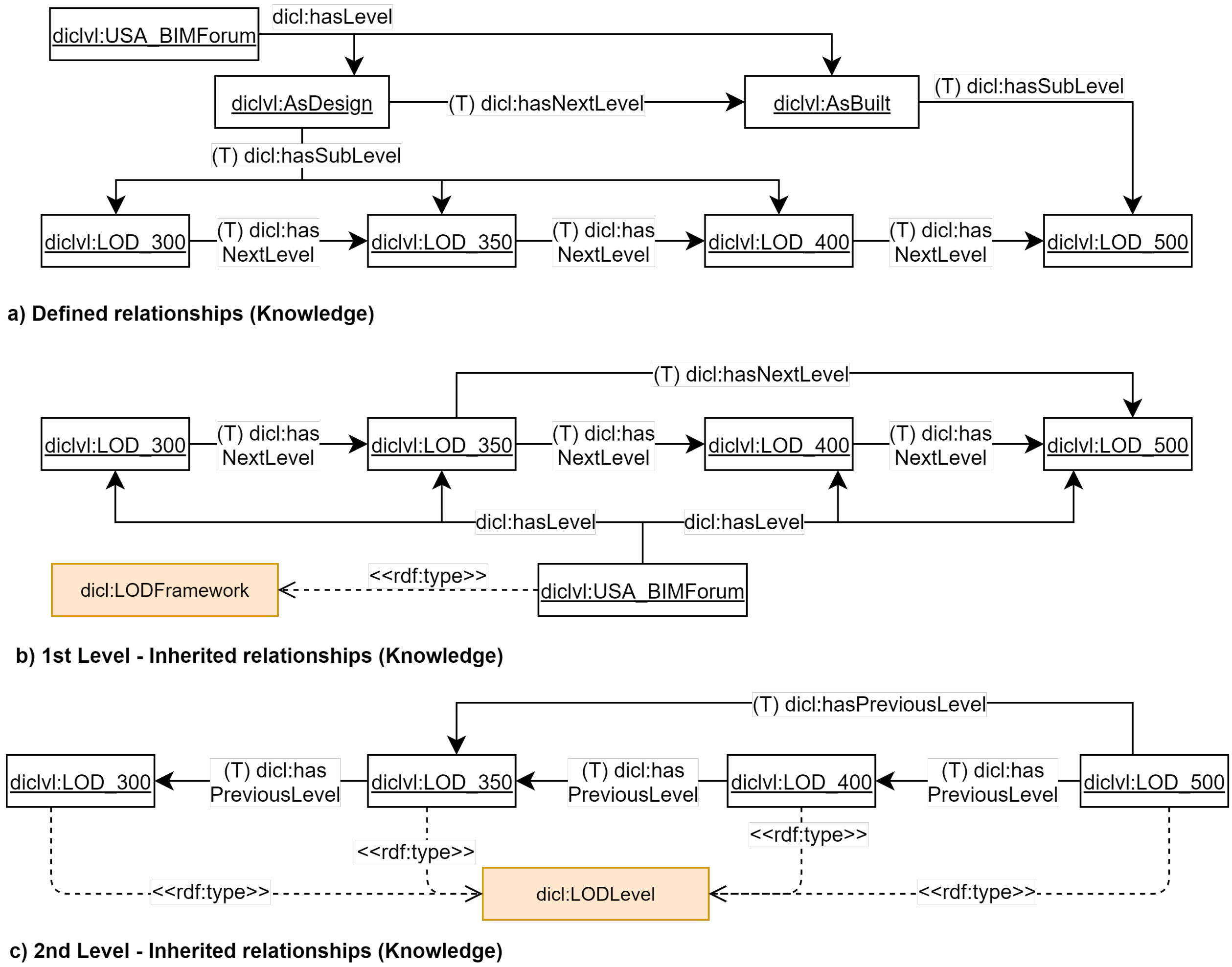


Figure 6: Relationship between the LOD scales

The relationship between the LOD levels is modelled by using transitive object property dicl:hasNextLevel (see Figure 6) according to their role in information growth representation. In addition to this defined relation between adjacent levels, dicl:hasNextLevel and dicl:hasPreviousStage relationships are generated between the non-adjacent levels, for example, between *diclvl:LOD_350* and *diclvl:LOD_500*. These relationships are generated because of the transitive property characteristic of dicl:hasNextstage and dicl:hasPreviousStage.

3.3 LOD - Verification

The verification of the developed ontology framework is performed by running SPARQL queries on the demonstrated example. These queries are majorly developed by considering the competency questions used for the ontology development process. The information related to instances and relations between them is populated to the developed ontology framework based on the analysis results comprised in Table 5.

After the population of instances, a reasoner (Pellet) was used to inference the new knowledge and to check the consistency and correctness of the developed ontology. One example query along with the results are illustrated in Table 7 and further examples are provided in Annex 1, Table 14.

Table 7: LOD - Queries and their results – Part 1

CQ2: What is the link between the LOD system and its levels?		
SPARQL Query	**Query results**	
`SELECT ?system ?levels WHERE { ?system dicl:hasLevel ?levels. Filter(?system = diclvl:USA_BIMForum) }ORDER BY ASC(?levels)`	**?system**	**?levels**
	USA_BIMForum	AsBuilt
	USA_BIMForum	AsDesigned
	USA_BIMForum	LOD_100
	USA_BIMForum	LOD_200
	USA_BIMForum	LOD_300
	USA_BIMForum	LOD_350
	USA_BIMForum	LOD_400
	USA_BIMForum	LOD_500

Further queries were executed to extract the generated information to check the consistency and quality (using Snap Sparql view in Protégé). For a clear picture of the developed A-Box knowledge, the same set of queries are used on the non-inferenced information and the comparison of the results are indicated in Annex I.

4 LOD-based BIM data Management

The section provides an analysis for the management of LOD-based BIM data. This analysis focuses on LOD-based representational methods for BIM data (also called LOD sensitive BIM). It also aims to describe the sources of this BIM data generation and the representation of their evolution and refinement. The analysis also considers the developed workflow ontologies (see D3.4) and the other BIM4EEB ontologies developed as part of Deliverable D3.2 (Material, Energy, Occupancy, etc.). The task mainly uses the classes and concepts from ontologies already developed in WP3 and introduces a LOD-based BIM data management workflow.

4.1 Approach Analysis

The representation of BIM data in a LOD-sensitive manner is based on the following approach:

(1) Identifiers (URIs) of entities are not LOD sensitive, and

(2) The statements about objects can be LOD sensitive.

That is, objects themselves are not associated with a specific LOD level but their property values can be. It is therefore not possible to ask which LOD an object belongs to because the object can simultaneously have properties belonging to many different LOD levels.

The representation of multiple LOD specific values requires an extended mapping (*object, property, level → value)* instead of normal mapping provided by RDF triples (*object, property → value*).

Such mapping can be implemented in different ways in RDF as reviewed in the Deliverable 3.1 (BIM4EEB-D3.1, 2022). The approaches can be divided into static ones that support a fixed set of levels – such as context-specific properties (e.g., :planned_<property>, :actual <property>), or context-specific namespaces (e.g., planned:<property>, actual:<property>) – and dynamic ones that support a variables set of levels – such as objectified properties (related to RDF reification), singleton properties, and named graphs.

Since the Digital Construction Lifecycle ontology (DICL) should support different LOD systems (for instance, as used in different countries) and even systems that contain an unspecified number of levels (such as the LOIN standard), only the dynamic approaches can reasonably be considered for a generic ontology. Of these, the most established ones are named graphs and objectified properties. Representations to support both of these approaches have been implemented in Digital Construction Ontologies and below their roles in the representation of LOD data is described.

4.1.1 Named Graphs

An *RDF database* can contain multiple *repositories*, each of whose contents is an *RDF Dataset*. An RDF Dataset is a *set of RDF graphs*, comprised of one *default graph* and an unlimited number of *named graphs*.

Each named graph is identified by its own URI. The possibility to store and use data from multiple different graphs creates an additional dimension to RDF. Instead of triples, the statements are in reality represented as quads in the form <graph, subject, predicate, object> (BIM4EEB 2021, D3.1). It is easy to see how the extended mapping mentioned above can be implemented with quads.

Named graphs can have properties of their own – that is, triples whose subject is the URI of the named graph. These properties can describe the named graph itself and the dataset contained in the named graph, e.g. the origin, creation time, maximum LOD level, and so on. They can be regarded as metadata of the contained data. Named graphs are an efficient mechanism to represent the metadata that is same for all the contained data. However, when the different triples in the dataset have different metadata, named graphs is not an appropriate representational mechanism.

Named graphs have been used in the Digital Construction Contexts ontology (DICC) whose purpose is to capture different information realms in renovation and construction projects, for example, the realms of planned and actual values (BIM4EEB Deliverable 3.4). The central concept in DICC a *context* (dicc:Context) that is used to indicate different information realms and that is associated with a named graph (through dicc:hasContent).

The contexts are further used in the Digital Construction Information ontology (DICI) as a superclass of ISO 19650 *information containers* (dici:InformationContainer). Since a *BIM model* (dici:BuildingInformationModel) is a subclass of the information container, it will be stored in its own named graph. Consequently, different BIM models will be in different named graphs and potentially also each major version or revision of the same model has its own named graph. (BIM4EEB Deliverable 3.4)

The actual approach how contexts and named graphs are used in each project depends on the manner that the ISO 19650 information model is utilized and what is the federation approach used.

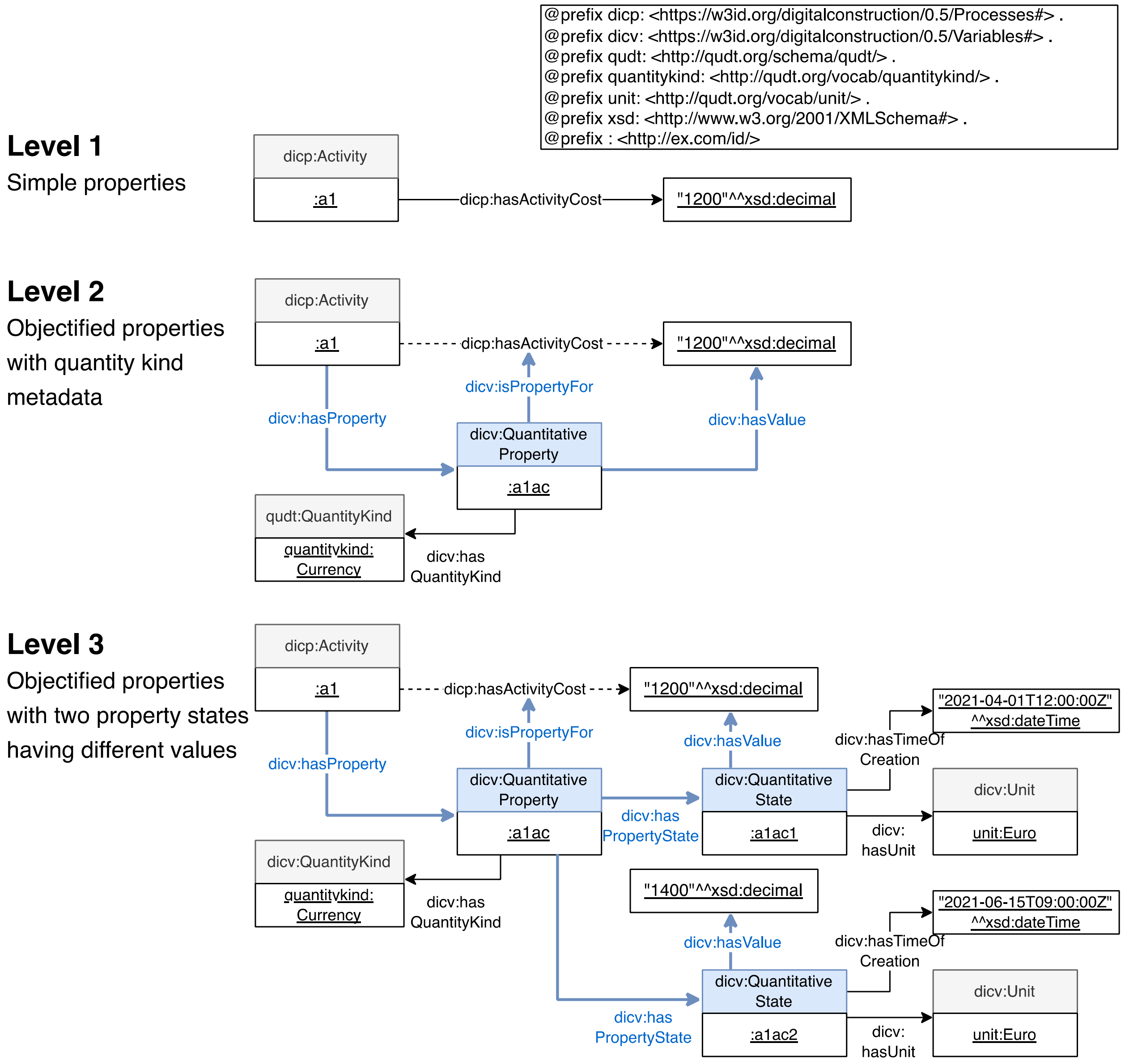


Figure 7: Different levels of property representation

4.1.2 Objectification of properties

Objectification of properties means that instead of triple holding the value itself, an intermediate object representing the property holds the value. This approach allows the flexible representation of an unlimited number of different additional properties for any simple property. This can be done both for

- *the property*, allowing the representation of property-related metadata, such as its quantity kind, constraints, etc., and
- *the values*, allowing the representation of multiple values in different property states, corresponding to LODs, time periods, together with value specific metadata, such as the source of the value, the time when the value was provided, and so on.

Property objectification is a well-known modelling mechanism supported by many established ontologies such as QUDT, SSN/SOSA or Saref. The Ontology for Property Management (OPM) (Bonduel 2018) specifies three different levels for representation of properties. These objectification model has been implemented in the Digital Construction Variables ontology (DICV) (BIM4EEB Deliverable 3.4) and shown and with the aid of examples in Figure 7. DICV is aligned with OPM (BIM4EEB Deliverable 3.6). OPM defines three levels of property representation:

- Level 1: Simple properties
- Level 2: Objectified properties
- Level 3: Objectified properties with multiple property states

The implementation of the levels based on DICV is illustrated in blue color in Figure 7. As can be seen, the approach used in Levels 2 and 3 makes it possible to add an unlimited number of additional attributes for each property (such as unit or quantity kind) and each property state (such as LODs, sources, time stamps, and so on). These capabilities would allow a complex LOD-sensitive representation of BIM data.

4.1.3 Adopted approach – conclusion

In the technical sense, the two mechanisms presented – named graphs and objectification of properties – are not in any conflict with each other. Named graphs works at the level of datasets (e.g., BIM models) and the property objectification at the level of individual datums in the dataset. It is thus technically possible to use them together and combine their strengths.

The approach adopted in BIM4EEB is based on the established practice that one BIM model can contain information at several different LOD levels. That is, all information in a model is not necessarily at the same LOD level. To capture these intra-model LOD distinctions, it is necessary to use objectified properties. However, at the more coarse level there can be different versions of models – perhaps documenting the situation at specific milestones or capturing a major revision, made necessary by a large design change. These major versions can be managed as different datasets.

In summary, major dataset-level revisions of BIM models are managed with named graphs through the information containers (dici:BuildingInformationModel) managed in the project information model (dici:ProjectInformationModel) and the actual management of LOD levels inside each of these models is addressed with objectified properties (dicv:Property, dicv:PropertyState). And the subsequent discussion will thus elaborate that approach.

4.2 Ontological representation

The ontology development process also progresses through addressing the fulfilment of requirements represented in natural language Competency Questions.

- **CQ5:** How is the BIM object is represented?

In the DICL ontology, the class dice:BuildingObject is used to represent the building object information. This class is related to the class dicv:Subject by an equivalent property.

- **CQ6:** How to represent multiple versions of information about the same object?

As clearly elaborated in the above section, an approach called Objectification of Properties is considered to represent the multiple versions of the same object's information. According to this approach, the defined classes dicv:Property and dicv:PropertyState make it possible to add an ultimate number of properties to building objects and their growth of accuracy throughout the project life-cycle. In detail, the object property is modelled by the class dicv:Property and the growth of the value of this property is comprehensively indicated in different property states using the class dicv:PropertyState. The relationship between these classes is represented by object property dicv:hasPropertyState. Additionally, these classes support the definition of meta-data attributes for each property (e.g. role or quantity kind, unit, value) and each property state (e.g. source, timestamp, value, unit, etc.). This property definition capabilities are attached to the building object using the object property dicv:hasProperty and its inverse property dicv:isPropertyOf.

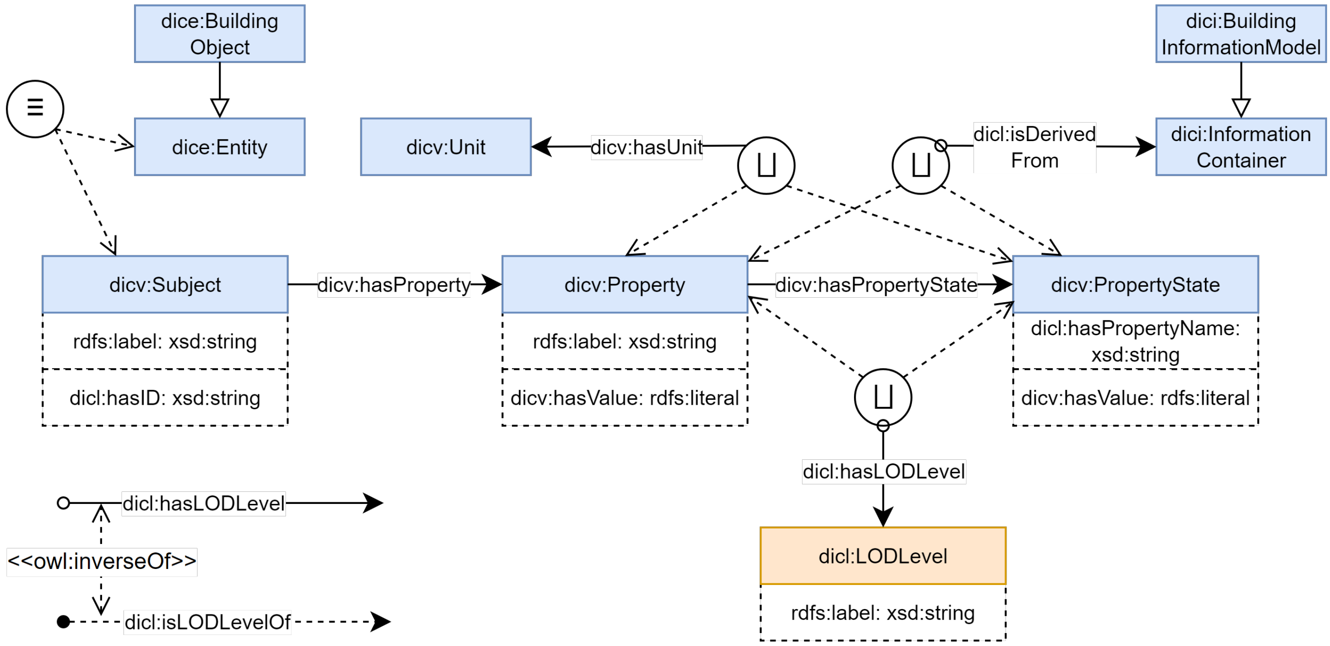


Figure 8: Building Product Data management

- **CQ7:** How the object properties and values for a specific LOD level are defined?

Once information objects are finished or new updates in information systems have been made, the relevant data can be converted into an RDF form and published as Linked Data to allow the integration of the relevant content with other renovation data. Each information object (explicitly or implicitly) provides information at certain LOD levels, and therefore the data should primarily be converted in a LOD sensitive manner. A difficulty that needs to be taken into account in such conversions is that the LOD frameworks currently used explicitly allow the same BIM model to contain data at different levels of detail. For instance, a structural model can contain a detailed design (e.g., at LOD400) of only one representative element of each type, while the other elements remain at the previous level of LOD (e.g., LOD350).

Within the DICL ontology development, this can be achieved by using the same approach called Objectification of the properties. This approach or specification also supports the modelling of explicit relationship between LOD and BIM attributes by using the object property dicl:hasLODLevel and its respective inverse property dicl:isLODLevelOf. This relationship makes it possible to represent the property value at its corresponding LOD level. Further relations between the LOD levels and their LOD structure are defined in the ontology representation detailed in Section 3 of this deliverable.

- **CQ8:** What are the sources for LOD data?

Along with the representation of LOD-sensitive BIM data, it is also necessary to represent the sources for this LOD data. The possible sources of LOD data are different kinds of information objects. In the context of BIM4EEB, these are primarily BIM models but can also include drawings, documents, messages and events/notifications. To address this requirement, a concept called dici:informationContainer is considered and liked to dicv:Property and dicv:PropertyState using the object property dicl:isDerivedFrom. This information container represents different data models, drawings, reports, etc.

The developed ontology approach is similar to the (Bonduel 2018) approach but a modification is considered to fulfil the needs of deliverable D3.3. which is the representation of LOD level as an owl:instanceOf dicl:LODLevel instead of the data property. This methodology allows users to directly represent the information with already defined levels, which are already categorized based on the LOD systems and contains relationships among the levels. This representation enables the data upgradations and query process much easier and reliable. Mostly eliminated human interpretation and textual errors in the data management process.

5 Renovation Process Modelling

The objective of this section is to develop an ontology representation for renovation process modelling which clearly describes the different activities, sequences, their interrelation with stakeholders involved and information requirements. Along with the ontology developments, the integration of the LOD-based BIM data representation in the renovation process modelling is also specified in this section. This integration mainly focuses on the reduction of the current gap between the process models and the developed BIM models. This integration task doesn't mainly depend on the development of a new ontology framework but to develop a set of required properties to indicate and manage LOD-based BIM data.

5.1 Renovation components

To achieve the objective, it is necessary to understand how the renovation workflow is implemented as part of the BIM4EEB project. A comprehensive analysis of the renovation workflow is presented in the deliverable D2.1 of Work Package 2. This section aimed to rewind the involved components in the renovation process described in D2.1. This process helps to understand the relation between these renovation components and supports ontology development.

5.1.1 Renovation activities

Activity is a task need to be carried out by a stakeholder or agent at a specific point of the renovation intervention. All these activities are listed as different renovation processes involved in the corresponding Building Lifecycle Stage (BLS). Approximately, 196 of such well-defined renovation activities are listed and explained in the deliverable D2.1, chapters 3 and 4, Tables 4-26. This complete workflow is majorly considered for the development of ontological representation for renovation processes. But due to confidentiality reasons, non of these activities are revealed in this deliverable. We agreed to denote these activities with dedicated numberings already assigned by D2.1 authors.

5.1.2 Stakeholders or agents

Stakeholders or the agents are the actors, who are involved in renovation activities. A sum of 38 stakeholders are represented in D2.1 and all are listed in also Table 2 of this deliverable. An analysis is performed on these stakeholders and their involvement in the activities categorized by the BLS. According to the defined set of renovation workflows (From Table 4 to 26 in D2.1), a relational based representation is generated between the Agents (stakeholders), Activities and Building lifecycle stages. This relationship is clearly indicated in Figure 9 and/or Figure 25. In Figure 9, these activities are represented with their respective numbering (red colour) and their representation in specific BLS is also denoted with numbering (Blue colour). For more information on BLS numbering, please refer to Table 1.

№	Stakeholders \ BLS	Building Lifecycle Stages (BS EN 16310)																			
		0		1			2					3		4					5		6	
		0.1	0.2	1.1	1.2	1.3	2.1	2.2	2.3	2.4	2.5	3.1	3.2	4.1	4.2	4.3	4.4	4.5	5.1	5.2	6.1	6.2
1	Inhabitant / End-user	1, 2		41		40						42	43	141, 142		154, 155, 156			168, 169, 170			
2	Client / Owner	3, 4, 5, 7	9	10, 11	15	52, 53													170, 171			182
3	Client' Adviser																					
4	Project Leader																					
5	Site Surveyor																					
6	Cost Consultant / QS				36																	
7	Health and Safety Adviser																				180	
8	Information Manager																					

Figure 9: Agent's involvement in activities w.r.t BLS

5.1.3 Activity sequence

According to Deliverable D2.1, the structure of the renovation processes are defined completely based on the BLS following the EN 16310. Figure 9 and Table 8 is also illustrating the same information. The details of all renovation activities at each stage of the overall renovation workflow are defined and detailed in Table 4 to 26 in D2.1. Similar to these definitions, to indicate the sequential appearance of activities, this deliverable also uses the BLS as a renovation activity categorization factor. Furthermore, developments in this process are presented in Section 5.4 of this deliverable.

5.1.4 Usecase

In this step, a set of simple use cases are defined with the aim of BIM4EEB tool requirements according to the renovation workflow described in D2.1. The idea of this implementation is to understand the involvement of various components (BLS, stakeholders, activities, etc,) and their relationship in the renovation process. We try to understand how do these components are used or at least related to other work package developments in future. Also, try to understand how does the development of DICL ontology should support domain-based process representations. With these use cases, we got a better understanding of the renovation process and used them as initial steps for the DICL ontology development process. We have outlined the three Use Cases as a dynamic combination of selected activities performed by related stakeholders during the renovation processes in the focus of the BIM4EEB. One is presented below and the remaining two are in Annex IV – Additional Use Case Definitions.

Use Case 1: Data Acquisition for HVAC Design, Operation and Efficiency Management

This use case describes activities related to data acquisition processes. All the identified activities may help in the development of digital tools for fast mapping of existing building data. The identified activities majorly provide the information regarding geometrical data, existing services systems, presently available heating and water pipes, ventilation systems, electrical cords, equipment, etc. Here, the numbers within the below table referring the activity numbers mentioned in renovation workflow developed in D2.1. The numbers used to represent BLS is completely based on the BLS numbering illustrated in Table 1.

Table 8: Stakeholders and their involvement in Use Case 1

BLS / Stakeholders	0	1	2.1	2.2	2.3	2.5	4	5	6
Technical Adviser									178
Project Leader		13, 14, 32	60, 61	73	95, 96				184, 185
Architectural Designer			55, 56	75		118			
Site Surveyor		16, 20, 21							
Client	3, 4								
Owner	3, 4	47, 48							
Local Authority				74					

However, we extended this analysis to the usecases defined in deliverable D3.1. The results from this analysis is presented below. The use cases defined in Deliverable 3.1 (Appendix I) are also related to the process steps (renovation activities) identified in Deliverable 2.1 (Table 4 to Table 12) and the LOD levels are related with each other. The information contents developed in the use cases are also indicated.

D3.1 Use cases		D2.1 Process steps	Developed information	Design LOD	
Construction management use cases				Min –	Max
CM1	Master planning	22-37, 64-66	Activities, WBS, precedences, schedules	LOD200	LOD400
CM2	Task planning	87-89, 97, 122	Activity flows: object, location, agent, equipment, material	LOD300	LOD400
CM3	Progress coordination	143-144	Sensor data, progress updates, issues	LOD300	LOD400
CM4	Plan revisioning	148, 162-163	Activities, WBS, precedences, schedules	LOD300	LOD400
CM5	Week planning	142	Task-level schedule, activated for execution	LOD400	
CM6	Stakeholder coordination	40-42, 141, 170	Alerts, messages, corrective measures	LOD300	
Energy-efficiency use cases					
EE1	Preliminary decision for the renovation	7	Cost-benefit estimate based on goals and historical data	LOD100	
EE2	Renovation project initiation	25	Feasibility indicators based on simple representative model	LOD100	
EE3	Identification of conceptual design alternatives	56	Renovation scenarios and measures based on arch BIM	LOD200	
EE4	Preliminary energy simulations of alternatives	79	Performance indicators (OPRs) of renovation scenarios	LOD200	
EE5	Detailed energy simulation with BIM models	100	Detailed energy performance	LOD300	
EE6	Detailed simulation based on a digital twin	124	Energy performance taking real-time data into account	LOD500	
EE7	Implementation of the renovation measures	142-154	Issues, changes, as-built data	LOD350	
EE8	Evaluation of performance at operations	167, 173	Energy performance taking real-time data into account	LOD500	
EE9	Recycling of the products and materials	184	Identification of recycles parts and materials from waste	LOD500	
Occupants behaviour and comfort use cases					
OB1	Establish comfort preserving framework	38, 54, 69	Occupant behavior and comfort parameters and targets	LOD100	
OB2	Ensure comfort conditions during construction	92, 113, 130	Design solution that ensure targets	LOD300	LOD400
OB3	Construct occupant comfort solutions	154-155	Issues, changes, as-built data	LOD400	
OB4	Maintain comfort preserving framework	168-170	Operation data, user feedback, issues, changes	LOD500	
Equipment and material use case					
EM1	Acquire preliminary material/equipment data	17-19	Overview of existing materials and products	LOD200	
EM2	Acquire detailed material and equipment data	20	Detailed properties of relevant materials and products	LOD300	
EM3	Interventions/schedules/costs of selections	23-36	Renovation scenarios, renovation measures, schedule, cost	LOD300	
EM4	Agree to share information	39, 45	Agreement between parties	LOD200	
EM5	Prepare the concept design with documentation	54-72	Concept designs of renovation scenarios	LOD100	
EM6	Integrate work from discipline designers	68	Integrated concept designs	LOD200	LOD500
EM7	Prepare preliminary design documentation	90-91	Material/equipment related design documentation	LOD200	
EM8	Prepare developed design	95-117	Developed designs of renovation scenarios	LOD200	
EM9	Prepare detailed design	118-138	Detailed designs of renovation scenarios	LOD300	
EM10	Oversee construction and commissioning	139-163	Sensor data, progress updates, issues, changes, approvals	LOD400	
EM11	Operate the building to satisfy owner/occupants	164-176	Operation data, user feedback, issues, changes	LOD500	
EM12	Preparation of end-of-live BIM model	177	BIM design focusing on recycled parts and information	LOD500	
Indoor air quality use cases					
IA1	Identification of IAQ parameters	17-19,23-25,3867-69,91-92	Existing IAQ conditions and occupant preferences	LOD100	
IA2	Establishment of a health/IAQ environment	111-113,127-130,154-155	IAQ target settings, desings satisfying the targets	LOD200	LOD500
IA3	Maintaining of a IAQ preserving framework	168-169, 172-173	IAQ data, performance indicator values, issues	LOD500	
Acoustics use cases					
AC1	Early identification of acoustical requirements	17-19, 23-25	Existing acoustic properties, regulations/requirements	LOD100	
AC2	Acoustical parameters for engineering	38, 67-69, 91-92, 111-113	Acoustic properties of elements and systems designed	LOD200	LOD500
AC3	Implementation of acoustic solutions	127-130, 154-155	Acoustic parameters for construction workers	LOD400	
AC4	Maintaining acoustical indoor conditions	168-169, 172-173	Real-time acoustic data, issues, corrective actions	LOD500	
Building performance use cases					
BP1	Documentation of User Comfort	17-18, 38, 168-169	Comfort parameters/thresholds from regulation/meas.	LOD100	LOD500
BP2	Documentation of System Usage	19, 164-167	Control systems data related to BIM	LOD100	LOD500
BP3	Documentation Energy Consumption	17, 167, 171	Metered consumption, weather conditions with time/BIM	LOD100	LOD500
BP4	Documentation of occupation density	12-18, 168-170	Presence data on location hierarchy from BIM	LOD100	LOD500
Cost modelling use cases					
CO1	Estimate renovation cost	5-9, 25-26	Relocation, demolition and renovation cost estimates	LOD100	
CO2	Specify approximate renovation cost	36, 49-51	Costs based on approx. quantities and average unit prices	LOD200	
CO3	Specify exact quantities and cost	60-63	Cost based on quantities and market prices for unit cost	LOD300	LOD400
CO4	Specify environmental/life-cycle cost	78-79, 100	Embodied energy and environmental impact	LOD500	

5.2 Representational Schema

Efficient management, collaboration and sharing of object information throughout the BLS and among stakeholders are fundamental to ensure the successful implementation of a project (Mangialardi et al. 2017). The management of a project's lifecycle information ensures the reduction of error-prone operations, data communication problems, and provides significant efficiency benefits, time-saving, etc. (Di Biccari et al. 2018). The major components involved in the renovation lifecycle data management are BLS, Activities within BLS, stakeholders, Level of detail, product data and use-case based exchange requirements. The optimized modelling of these components can enhance the collaboration benefits through the effective and iterative exchange of information (Karlapudi et al. 2020).

Based on the analysis of renovation components, renovation workflow and LOD-based BIM data management, a representational schema for Renovation (Product-Process) Modelling is presented in Figure 10. The development goal is to clearly describe the relationship between use cases, activities, stakeholders involved, the required information (BIM-objects and properties), BLS, and LOD.

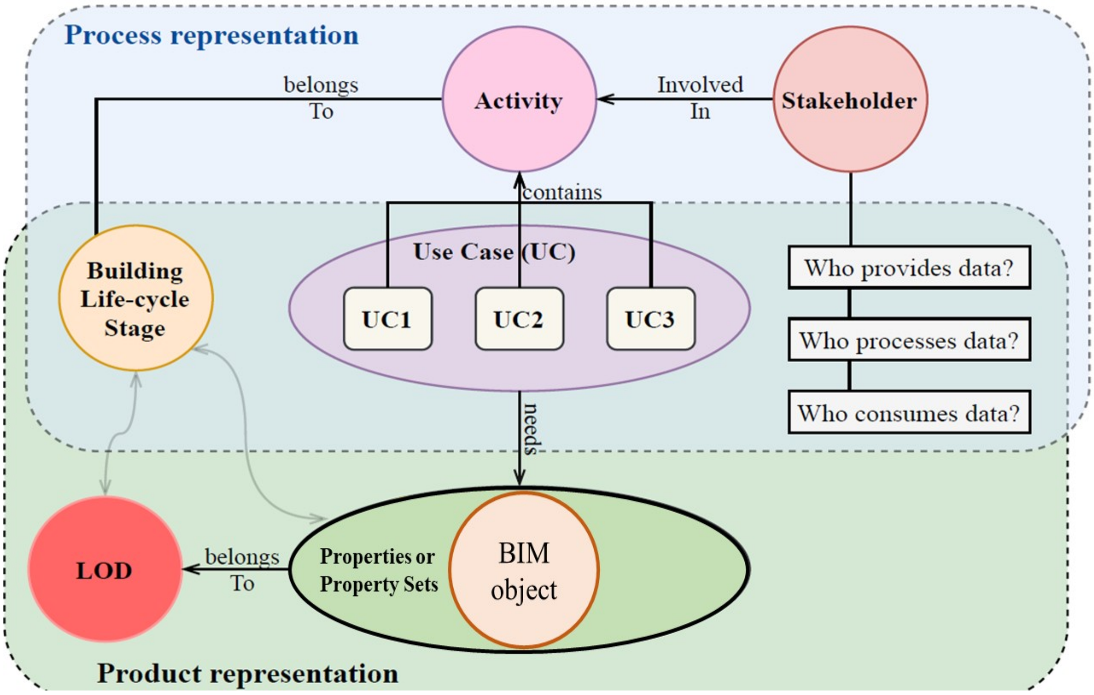


Figure 10: Representational schema - Renovation Product-Process representation

The Product representation indicates the relation between BIM objects and their level of detail needed in Building Lifecycle stages through the involved activities in it. Through the analysis of specified renovation processes defined in D2.1, a relationship between BLS and BIM entities is recorded as Entities-BLS-matrix represented in Figure 24 of Annex III – Integrated Product-Process modelling examples. Based on the analysis and previous experience from the partners, this figure also tried to explain the needed level of detail for the BIM data in a specific BLS to complete the required renovation activities.

Similarly, the Process representation describes the role of different stakeholders involved in renovation activities listed in various Building Lifecycle Stages. The relationship between these components is already analysed and represented in Figure 9. Further explanations regarding the ontology development process are comprehensively elaborated within the coming section.

5.3 Ontological representation of renovation workflow

The work involved in this section is to develop an ontological representation of renovation processes and link it to the LOD-based BIM data. Similar to the above sections, the concept of CQs is used for requirements specification for renovation process ontology development. The ontology development process progresses through addressing these requirements or CQs.

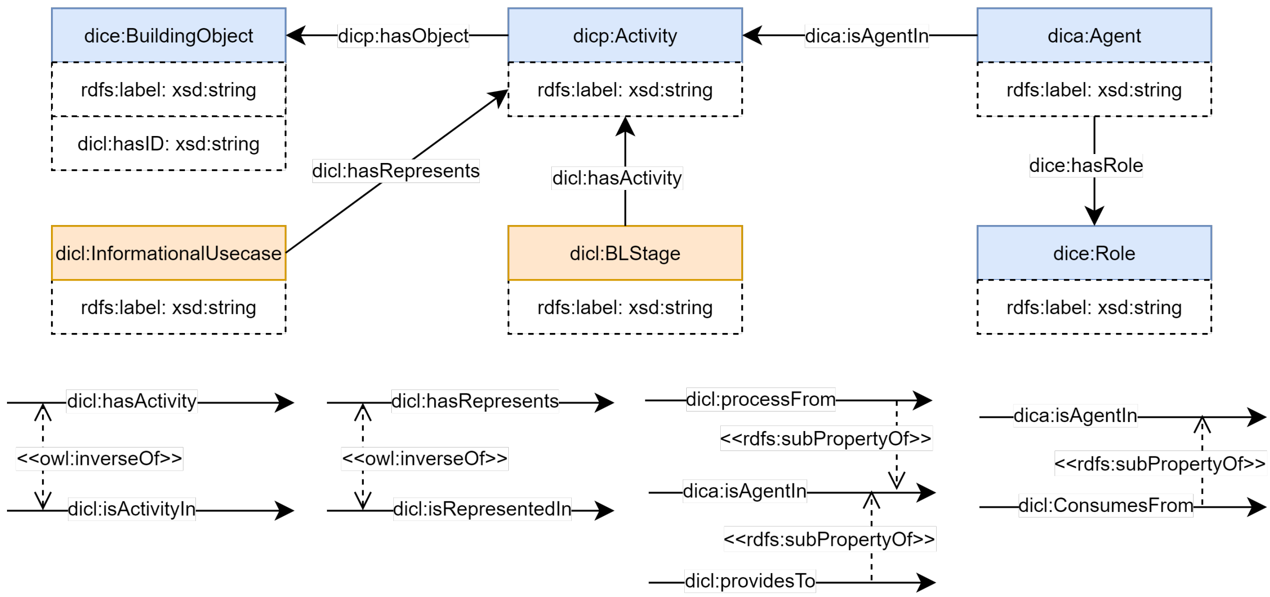


Figure 11: Renovation Workflow – Ontology representation

- **CQ9:** How the activities are defined in the renovation workflow?

According to the provided discussion in Section 5.1.1 of this deliverable, different renovation processes are defined as activities within the renovation workflow. Within the DICL ontology, these activities are represented with the class dicp:Activity. All these activities are listed or structured according to the BLS stages defined from EN 16310. A class dicl:BLStage is adopted to indicate different Building Lifecycle Stages from different standards and publications. The representation of these activities in a specific BLS is achieved by generating the relationship between the classes dicp:Activity and dicl:BLStage by using object property dicl:hasActivity and its inverse property dicl:isActivityIn.

- **CQ10:** How to identify the sequence of activities?

As described in D2.1, renovation workflow, the sequential relationship between these activities are represented based on the BLS. In the renovation workflow, the activities are divided or categorized completely based on the structure of BLS. In technical terms, the progressive growth of renovation interventions can easily be represented or tracked through the BLS. Also, the relation between these different stages can easily be represented in ontology structure. Based on these advantages, in DICL ontology, the same BLS concept is adopted to represent the sequence of renovation activities.

Furthermore, the comprehensive and elaborated explanations regarding the ontological representation of BLS frameworks are presented in Section 5.4 of this deliverable.

- **CQ11**: How are the stakeholders or agents related to activities?

An intensive analysis of the relationship between the agents and activities are already presented in subsection 5.1.2 of this deliverable. Based on these analyses, In the DICl ontology, the class dica:Agent is adopted to specify all the actors/stakeholders involved in the building renovation process. Also, a class dice:Role is adopted in DICL ontology to assign the role to an agent based on their involvement in the specific activity. As represented in Figure 11, the list of identified roles for the agents are instances of the class dice:Role. These specific roles are assigned to agents through the object property dice:hasRole and its inverse property dice:isRoleOf.

Apart from the representation of the relationship between the activities and the stakeholders, an analysis is carried on the participating role of the stakeholders in each activity. This analysis was carried out based on the defined workflows in Deliverable D2.1 (Table 4 to 26). These analysis results are represented in tabular format in Figure 12 and/or Figure 26. From this analysis, 3 information roles are identified for the agents. These roles are (1) Information Provider, (2) Information Processor and (3) Information Consumer.

Activity №	Stakeholder's №		
	Information Provider	Information Processor	Information Consumer
1	1	2	16
2	1	2	2
3	2	2	2
4	2	2	2
5	2	3	3

Figure 12: Information flow roles within the activities

To accommodate these additional roles in DICL ontology, a class dicl:informationRole is defined as a subclass to dice:Role to substantially accommodate different roles, such are dicl:InformationConsumer, dicl:InformationProcessor, dicl:InformationProvider.

Similarly, a simple relationship is defined between activity and the agent by using the object property dica:hasAgent and its respective inverse property dica:isAgentIn. Furthermore, the object property dica:isAgentIn is categorized into three sub-properties dicl:consumesFrom, dicl:providesTo and dicl:processFrom, which are used based on the role of the agent.

- **CQ12:** How does the renovation process linked to BIM data represented in LOD-sensitive manner?

As illustrated in Figure 10, the connection of LODs to renovation processes is generated based on the information requirements (BIM data). For example, different activities require specific information regarding the building objects available at a needed level of detail (LOD). In this DICL ontological framework, an explicit mapping methodology is specified to represent these requirements. This methodology specifies the following information for each activity.

- Object (dice:BuildingObject);
- Property name (or a property set or other grouping of properties);
- Nature of data needed (e.g., a LOD level or other specification).

A difficult representational challenge is how to represent the values of the same properties of objects at multiple different levels. The most typical approach is to use the objectification of properties, which was explained in Section 4.1. By using this approach, an ontological representation of LOD-based BIM data on the attribute or property level is achieved and elaborated in Section 4.2.

In Figure 11, this link between activities and LOD-sensitive BIM data is indicated by using the object property dicp:hasObject and its inverse property dicp:isObjectIn. This relationship allows inheriting the required level of data for each renovation activity within the building lifecycle stages.

- **CQ13:** How to represent specific use cases within the renovation workflow?

Atomic activities can be grouped forming larger processes or even use cases. Thus Use Cases can be defined as a special type of activity. The specification of Use Cases may help to simplify navigation through the knowledge space. It also provides an instrument to provide access to information requirements of specific tools or domain-specific processes involved in the renovation process. Figure 13 shows, how activities that were defined in D2.1 are grouped into larger activities called Use Cases.

Use Cases Names	Abbreviation	Activities supporting Use Cases 1 - 3												Use Cases Definition
Use Case 1	UC1	3, 4	13, 14, 16, 20, 21, 32, 47, 48	55, 56, 60, 61	73, 74, 75	95, 96		118			-	-	178, 184, 185	Data Acquisition for HVAC Design, Operation and Efficiency Management
Use Case 2	UC2	-	17, 18, 19, 27	63-68	76-79, 83, 84, 86-91	97, 98, 99, 100, 101, 102, 103, 104, 105, 107-110		119, 120, 121, 122–129, 131, 132, 134, 135			141, 142, 160, 161	-	181, 182, 183, 187, 191, 195	HVAC Design, Operation and Efficiency Mgmt. for Fast Track Renovation Operation
Use Case 3	UC3	-	22, 28, (29, 30), 33, 34, 35, 37, 40, 44			111, 112, 116, 117					143, 144, 145, 146, 147, 148, 155, 156, 159	166, 167, 168, 169, 170, 173, 174, 175	189, 192	Fast Track Renovation Operation

Figure 13: Dynamic allocation of Activities to Use Cases

Concerning the DICL ontology (see Figure 11), this grouping mechanism is represented within the ontology by a class dicl:InformationalUsecase. The involved activities (dice:Activity) within this use case are represented by using an object property called dicl:hasRepresents and its inverse property dicl:isRepresentedIn.

5.4 Building Lifecycle Stages (BLS)

In common practice, activities are classified according to BLS. Deliverable D2.1 provides an extensive analysis of activities for building renovation in each Building Lifecycle Stage. The relationship between BLS and the Activities is represented in the DICL ontology framework with the help of object property dicl:hasActivity and its inverse property dicl:isActivityIn. It is illustrated in Figure 11. Also in DICL ontology development process, the Building Lifecycle Stage is considered as a criterion for defining the sequential relationship between the activities through the definition of the relationship between stages.

However, the activity analysis in Deliverable D2.1 primarily considers BS EN 16310:2013. In practice there exist many other systems or frameworks to represent these lifecycle stages. Thus, the development of the ontological representation for BLS should be capable to accommodate these different specifications. This section aims to investigate alternative BLS representations. Table 9 provides an overview of specifications provided in EN 15463 Sustainability of construction works, BS EN 16310:2013 – Engineering services terminology to describe engineering services for buildings, infrastructure and industrial facilities (BS EN 16310:2013), HOAI – Official Scale of Fees for Services by Architects and Engineers (HOAI), RIBA Plan of Work (RIBA 2020), and ISO 22263:2008 – Organization of information about construction work – Framework for the management of project information (ISO 22263:2008).

Table 9: BLS in different standards and publications

Standards / Stages	EN 15463		BS EN 16310:2013		HOAI	RIBA	ISO 22263
1	Before Use Stage	Product Stage	Initiative	Market study Business case	Establish the base of the project	Strategic Definition	Inception
2			Initiation	Project initiation Feasibility study Project definition		Preparation and Brief	Brief
3			Design	Conceptual Design Preliminary Design Developed Design (B&I) Technical Design Detailed Design	Preliminary Design Final Design Building Execution Drawings Preparation of contract award	Concept Design Developed Design Technical Design	Design
4					Building permission applications		
5			Procurement (IF)	Procurement Construction Contracting	Assisting the award process		
6		Construction Stage	Construction	Pre-construction Construction Commissioning Hand Over Regulatory Approval	Project Supervision (Construction Supervision) Project control and documentation	Construction Handover and Close Out	Production
7	Use Stage		Use	Operation Maintenance		In Use	
8	End of Life Stage		End of Life	Revamping Dismantling			Demolition

5.4.1 BLS - Ontology representation

The application or the usage of BLS in building renovation projects is based on the locality, requirements, legislation, etc. Thus, a newly proposed ontological schema for a BLS should be capable to represent these different standards. Similar to the LOD framework, an ontological representation for the BLS framework is proposed to denote different lifecycle stages as shown in Figure 14. As similar to the previous ontology development process, this development of ontology class and relationships to represent BLS is progressed based on the concept of Competency Questions (CQ). The development process is further explained below.

- **CQ14:** How to enable the representation of multiple BLS systems and/or stages?

The adopted modelling methodology is similar to the LOD approach elaborated in Section 3.1. In DICL ontology, the classes dicl:BLSFramework and dicl:BLStage are defined to represent various BLS systems and stages respectively. Various BLS systems can be defined as instances to the class dicl:BLSFramework and be assigned to the processes based on the project requirements.

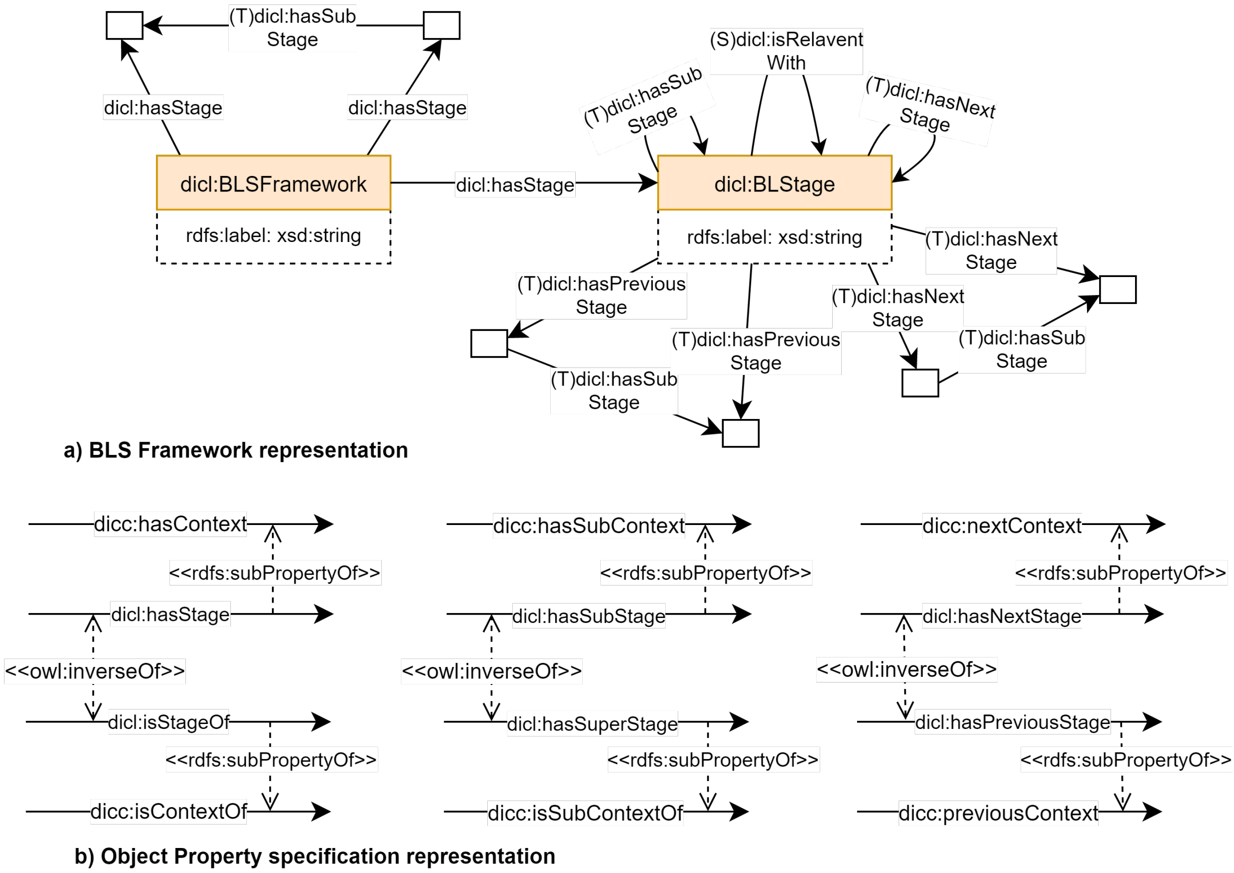


Figure 14: BLS in ontology representation

- **CQ15:** What is the link between the BLS system and its respective Stages?

Similar to the various BLS system definition, stages of the respective BLS system are defined as instances to dicl:BLSStage. The relationship between these stages to the BLS system is generated with the help of object property dicl:hasStage and its inverse property dicl:isStageOf.

As represented in Table 9, the stages from BS EN 16310:2013 are represented as main-stages and sub-stages to respective main-stage. The relationship between the main-stage and its respective sub-stages are defined by using the object property dicl:hasSubStage and its inverse property dicl:hasSuperStage. To represent the relationship between the BLS system to these sub-stages is generated by using the subProperty chain axiom (dicl:hasStage o dicl:hasSubStage → dicl:hasStage) defined to the object property dicl:hasStage. Figure 14 is clearly illustrating these relationships along with the defined axioms.

- **CQ16:** What is the relation between stages?

The relationship between the stages are defined by using the object properties called dicl:hasSubStage, dicl:hasNextStage and their respective inverse properties dicl:hasSuperStage, dicl:hasPreviousStage. Similarly, the transitive character of the properties is enabled by assigning owl:TransitiveProperty to these object property which further enables the aggregated relationships between the stages.

- **CQ17:** What is the relation between the main-stage to sub-stage of other main-stage?

The complex part of this ontology development process is to generate the relationship between the main-stage and sub-stages from another main-stage according to the BS EN 16310:2013. For example, as represented in Table 9, the human can easily interpret that the sub-stage “Pre-construction” is the next stage after the main-stage “Design”. These complex relations are modelled in the ontology by using axioms dicl:hasNextStage o dicl:hasSubStage → dicl:hasNextStage, dicl:hasPreviousStage o dicl:hasSubStage → dicl:hasPreviousStage assigned to properties dicl:hasNextStage, dicl:hasPreviousStage respectively.

- **CQ18:** How to represent the mapping between different BLS systems?

In DICL ontology, the mapping relationships between the stages from different BLS frameworks are modelled by using the symmetric object property dicl:isRelaventWith.

To provide a better understanding of this developed approach, an example demonstration is generated for the BLS system - BS EN 16310:2013 and elaborated clearly in the below section.

5.4.2 Demonstration of ontology approach

For a clear understanding of the functionality of this ontology framework, a demonstration is considered. For the demonstration, the BLS representations from the standard (BS EN 16310:2013) illustrated in Table 9 is considered. The list of stages from (BS EN 16310:2013) is represented as main-stages and a couple of sub-stages (see Figure 15 and Figure 16).

As represented in Figure 15, the instances *dicstg:Initiative*, *dicstg:Initiation* and *dicstg:Design* are assigned as stages for the instance *dicstg:BS_EN_16310* using dicl:hasStage object property. Since the object property's (dicl:hasStage) domain and range are fixed to the classes dicl:BLSFramework and dicl:BLStage respectively, the inferencing engine automatically develops new knowledge by concluding the instance *dicstg:BS_EN_16310* belongs to the class dicl:BLSFramework and the other instances are belonging to the class dicl:BLStage.

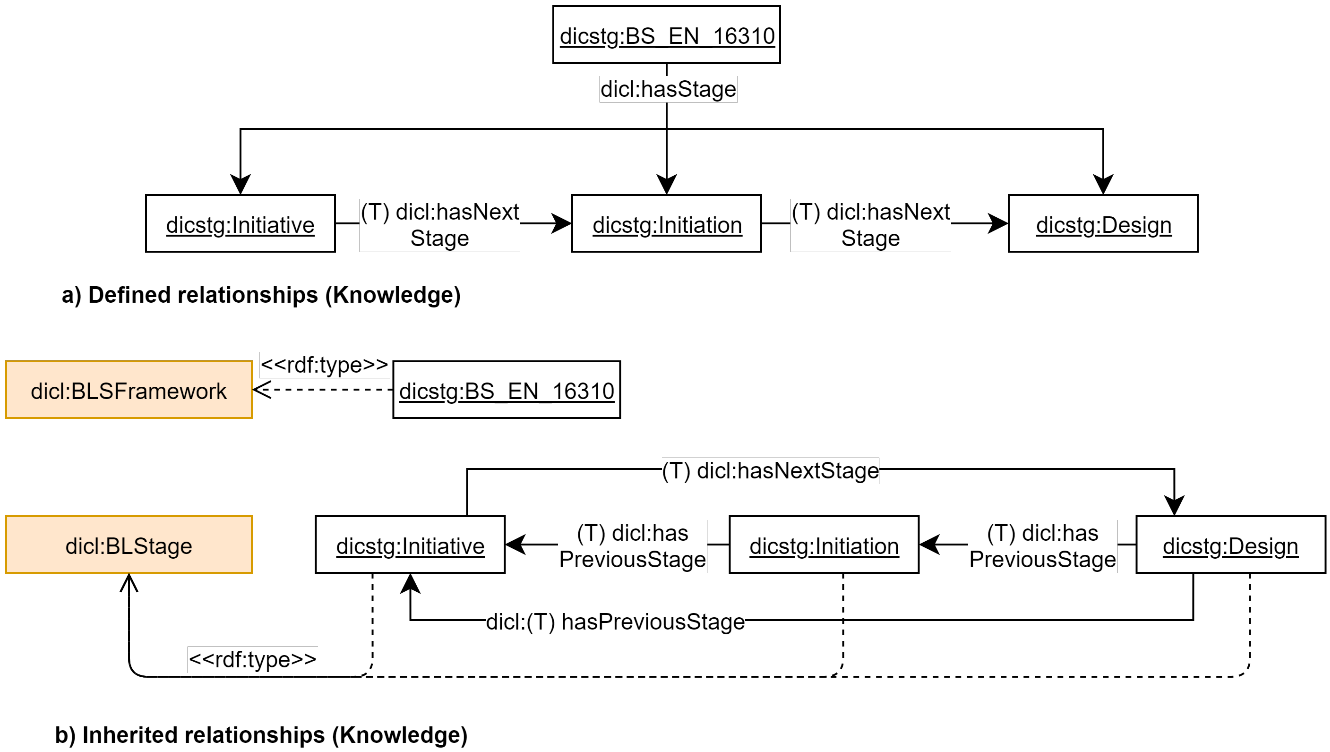


Figure 15: Demo1 – Relation between main stages of (BS EN 16310:2013) BLS framework

Furthermore, the relationship between the neighbour instances *dicstg:Initiative*, *dicstg:Initiation* (and *dicstg:Initiation, dicstg:Design)* are defined using dicl:hasNextstage transitive object property. As defined in the ontology framework, the dicl:hasNextstage object property has an inverse relationship dicl:hasPreviousStage. Because of this definition, the dicl:hasPreviousStage relationship is inherited between these instances. Along with this inherited information also new relationships are generated between *dicstg:Initiative* and *dicstg:Design,* which is clearly represented in Figure 15. These generated relationships are because of the transitive property characteristic of the object properties dicl:hasNextstage and dicl:hasPreviousStage.

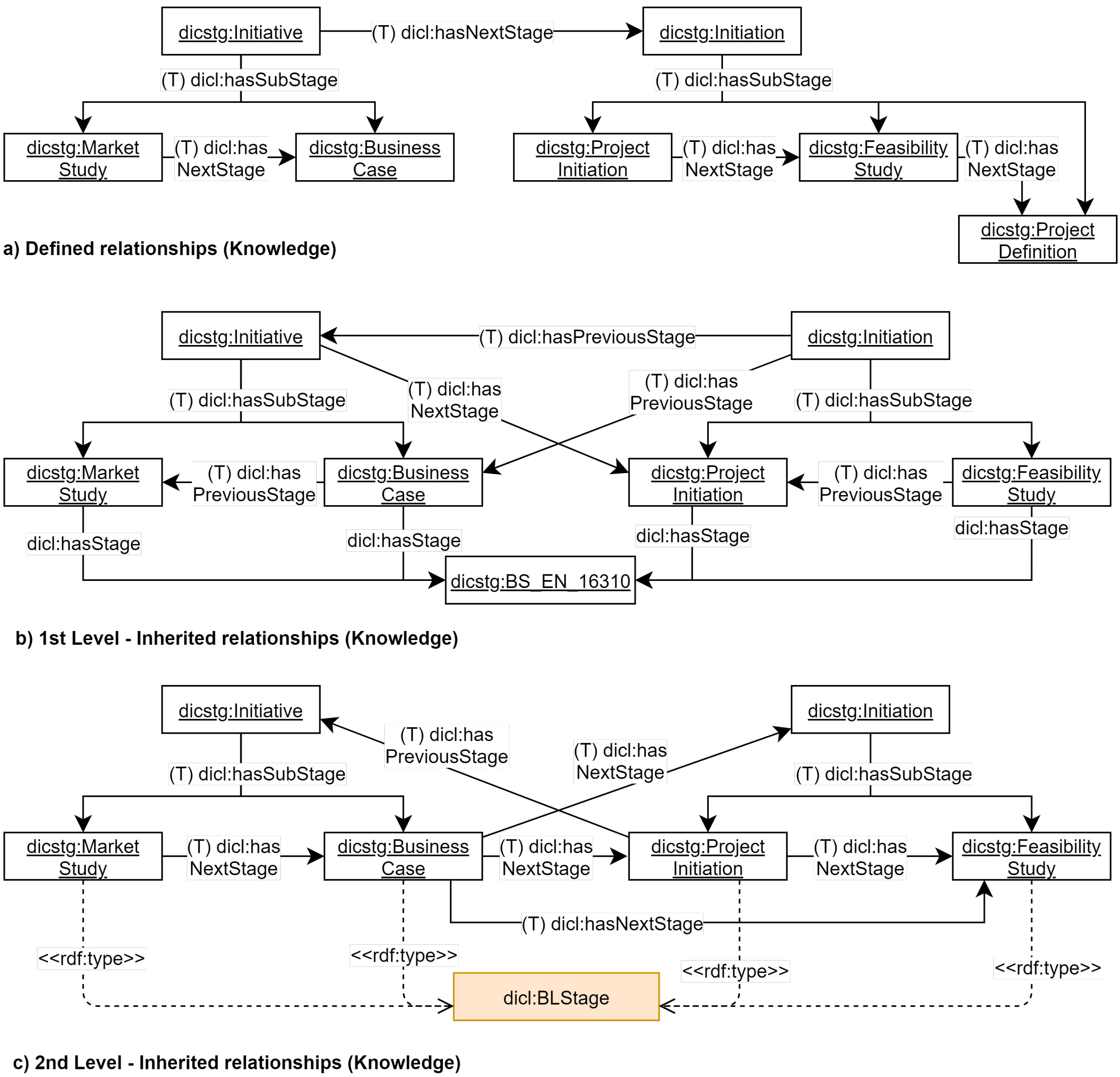


Figure 16: Demo2 – Relation between sub-stages of (BS EN 16310 2013) BLS framework

Sub-stage information regarding each main-stage is defined by using the transverse object property called dicl:hasSubStage and its inverse property dicl:isSuperStage. Because of the transverse nature of these object properties, further sub-levels of sub-stage are considered as sub-stages to the main-stage. Similarly, due to the defined axiom to dicl:hasStage property, all these sub-stages are inferred as stages (dicl:BLStages) to the dicl:BLSframework (*dicstg:BS EN 16310*).

Furthermore, the relationship between the sub-stages of a mainstage is developed by using dicl:hasNextstage. Due to this defined relationship, the inferences relations from its inverse property and the transitive property characteristic are illustrated in Figure 16.

But the complexity is to infer the relationship between the main-stage (*dicstg:Initiative*) and a sub-stage (*dicstg:ProjectInitiation*) of another mainstage (*dicstg:Initiation*), similarly between the sub-stage (*dicstg:BusinessCase*) of a mainstage (*dicstg:Initiative*) and the sub-stage (*dicstg:ProjectInitiation*) of another mainstage (*dicstg:Initiation*). This complexity of inferencing the new knowledge is resolved eventually by creating property axioms to dicl:hasNextstage and dicl:hasPreviousStage as represented in Figure 14.

The understanding of the inference relationship between the main stage and the sub-stages of another main stage is pretty straightforward from the assigned axioms but the generated links between sub-stages of different mainstages are quite complex to understand.

The assigned axiom to dicl:hasNextstage generates the next-stage relationship between *dicstg:Initiative and dicstg:ProjectInitiation.* It also means that *dicstg:Initiative* is a previous-stage to *dicstg:ProjectInitiation* and this information is generated based on the inverse property of dicl:hasNextstage, which is dicl:hasPreviousStage.

Now the axioms assigned to dicl:hasPreviousStage further inherits the previous-stage relationship between *dicstg:ProjectInitiation and dicstg:BusinessCase.* These inherited relationships are clearly represented and distinguished within Figure 16.

5.4.3 Verification of ontology approach

Similar to the LOD framework validation, the same methodology is adopted to validate the BLS framework. The validation of the developed framework is performed by running the set of SPARQL queries on the demonstrated example. As represented in Table 9, the information related to instances and relations is developed to the main default framework ontology.

After the population of A-Box instances, a reasoner (Pellet) was used to develop the new knowledge and to check the consistency and correctness of the developed ontology.

Thereafter, several queries were performed (using Snap Sparql view in Protege) to extract the generated information and to check the consistency and quality by comparing query results against the original information from BLS standards.

A set of simple queries and their results regarding the relationship between the stages is presented in the below table. This table represents the consolidated results from the Snap SPARQL query engine. For a clear picture of the developed A-Box knowledge, the same set of queries are used on the non-inferences information and the comparison of the results are presented in Annex II.

Table 10: BLS - Queries and their results – Part 1

<table>
<tr><td colspan="4">CQ16: What is the relation between stages?
CQ16.1:- What are the next stages of the stage “Construction” according to BS EN 16310?</td></tr>
<tr><th rowspan="2">SPARQL query</th><th colspan="3">Query Results</th></tr>
<tr><th>?Framework</th><th>?Stage</th><th>?NextStages</th></tr>
<tr><td rowspan="6">SELECT ?Framework ?Stage ?NextStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Framework=dicstg:BS_EN_16310) .
Filter(?Stage=dicstg:4.BS_EN_Construction) .
?Stage dicl:hasNextStage ?NextStages .}
ORDER BY ASC(?NextStages)</td><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>5.BS_EN_Use</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>5.1.BS_EN_Operation</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>5.2.BS_EN_Maintenance</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>6.BS_EN_End_of_Life</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>6.1.BS_EN_Revamping</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>6.2.BS_EN_Dismantling</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">CQ16.2:- What are the previous stages of the stage “Construction” according to BS EN 16310?</td></tr>
<tr><td rowspan="17">SELECT ?Framework ?Stage ?PreviousStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Framework=dicstg:BS_EN_16310) .
Filter(?Stage=dicstg:4.BS_EN_Construction) .
?Stage dicl:hasPreviousStage ?PreviousStages .
}
ORDER BY ASC(?PreviousStages)</td><th>?Framework</th><th>?Stage</th><th>?PreviousStages</th></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>0.BS_EN_Initiative</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>0.1.BS_EN_Market_study</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>0.2.BS_EN_Business_case</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>1.BS_EN_Initiation</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>1.1.BS_EN_Project_initiation</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>1.2.BS_EN_Feasibility_study</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>1.3.BS_EN_Project_definition</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>2.BS_EN_Design</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>2.1.BS_EN_Conceptual_design</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>2.2.BS_EN_Preliminary_design</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>2.3.BS_EN_Developed_design</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>2.4.BS_EN_Technical_design</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>2.5.BS_EN_Detailed_design</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>3.BS_EN_Procurement</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>3.1.BS_EN_Procurement</td></tr>
<tr><td>BS_EN_16310</td><td>4.BS_EN_Construction</td><td>3.2.BS_EN_Construction_contracting</td></tr>
</table>

6 Ontology Management & Validation

The section describes the derived alignment components between the Digital Construction Lifecycle (DICL) ontology and the external and/or BIM4EEB ontologies. The concept of Horizontal and Vertical segmentation is adopted to represent these alignments or dependencies between the ontologies. More specifically, these alignment concepts are comprehensively explored as part of the evaluation and validation process planned in Task T3.6 of WP3.

6.1 External ontology alignment components

During the second cycle of Digital Construction Lifecycle (DICL) Ontology development, the usage of classes from other (related) BIM4EEB ontologies was optimised. One of the reasons is to reduce the redundancy and the repetition of similar concepts within the BIM4EEB ontology suite.

The horizontal and vertical segmentation of these classes is not discussed as part of this deliverable. Instead, this is part of the discussion in deliverable D3.6. This section is dedicated to focusing on the management of the Building Lifecycle and BLS ontologies.

Table 11: External ontologies aligned with DICL ontology

Ontology	Prefix	Namespace
Organization ontology	`org`	`http://www.w3.org/ns/org#`
ifcOWL ontology	`ifcOWL`	`https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL`

6.1.1 Direct alignment with External ontologies

The sub-section describes the alignment of external ontology components with DICL ontology components. The identified alignment components are presented in the below table.

Table 12: Alignment components with external ontology classes

DICL class	Relationship	External Ontology class
dicl:InformationFlowRole	rdfs:subClassOf	ifcOWL:ActorRole, org:Role
dicl:InformationConsumer	rdfs:subClassOf	ifcOWL:ActorRole, org:Role
dicl:InformationProcessor	rdfs:subClassOf	ifcOWL:ActorRole, org:Role
dicl:InformatioProvider	rdfs:subClassOf	ifcOWL:ActorRole, org:Role

6.1.2 Indirect alignment with External Ontologies

As said above, the Digital Construction Lifecycle ontology is majorly developed by using the components or classes from other BIM4EEB ontologies. In technical, these BIM4EEB classes are also been dependent or aligned with corresponding external ontology components. Which subsequently results in indirect relations or alignments of components of the Digital Construction Lifecycle ontology to external ontologies. This document does not specify those alignment concepts in detail.

However, we provide in Figure 17 a schematic describing the dependencies of ontologies in the BIM4EEB ontology suite (see also D3.4 and D3.6). Figure 17 indicates that the DICL Ontology (DICL) imports classes from Information, Agents, Processes, Entities, Contexts and Variables ontologies.

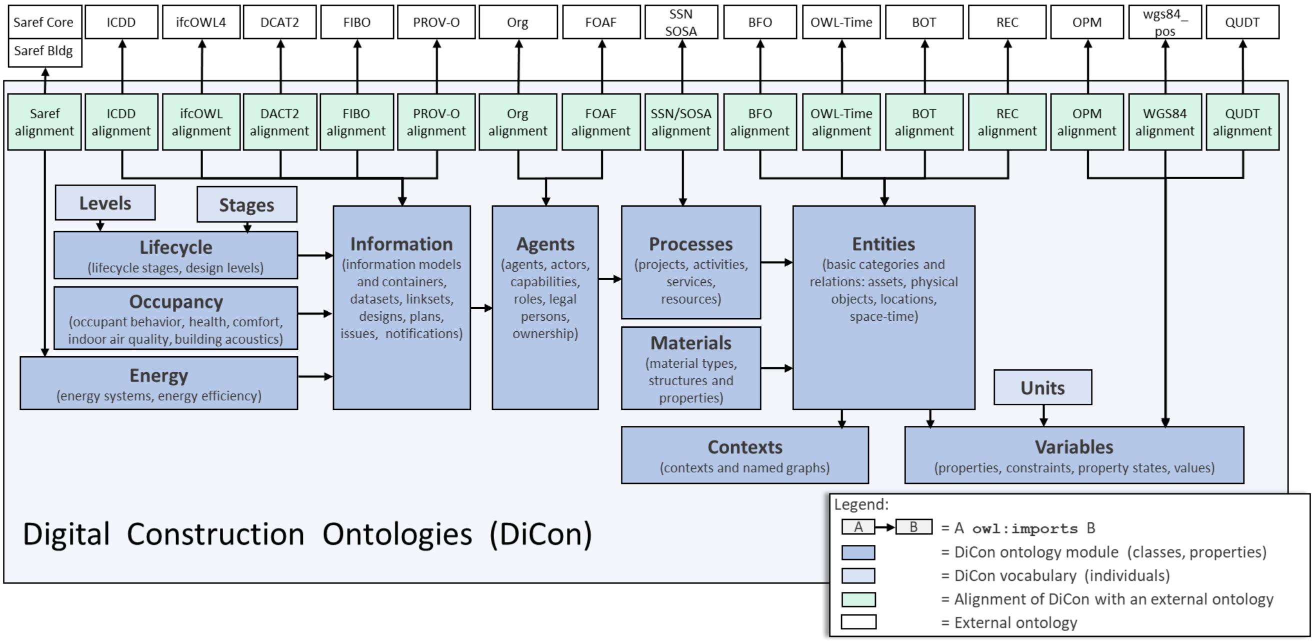


Figure 17: Dependency Schema for BIM4EEB ontologies

6.2 Dependency on BIM4EEB ontologies

Finally, Table 13 provides more detailed examples describing the dependencies of selected DICL-classes to complement classes of the BIM4EEB ontology suite. In general, LOD and BLS concept highly dependent on Context ontology from BIM4EEB ontology suite. The Context Ontology aimed to represent different contexts of BIM4EEB project and LOD is one of such contexts.

Table 13: Alignment components with BIM4EEB ontology classes

DICL class	Relationship	BIM4EEB Ontology class
dicl:LODLevel	rdfs:subClassOf	dicc:Context
dicl:BLStage	rdfs:subClassOf	dicc:Context
dicl:LODFramework	rdfs:subClassOf	dicc:ContextFramework
dicl:BLSFramework	rdfs:subClassOf	dicc:ContextFramework
dicl:InformationFlowRole	rdfs:subClassOf	dice:Role
dicl:InformationConsumer	rdfs:subClassOf	dice:Role
dicl:InformationProcessor	rdfs:subClassOf	dice:Role
dicl:InformatioProvider	rdfs:subClassOf	dice:Role

6.3 Validation of the ontology

The validation of the Digital Construction Ontology will be presented in Deliverable 3.6 according to the original plan. It will address the competency questions, their formalization in SPARQL according to the terms of DiCon and evaluation with test data. This Deliverable also provides some validation examples through the sample demonstrations to support the ontology development process.

7 Summary and Conclusion

The development goal of this deliverable is a Digital Construction Lifecycle (DICL) ontology to represent BIM data in different levels of detail (LOD) corresponding to renovation workflow. Intending to fulfil this objective, the deliverable starts with an extensive analysis of LOD systems defined in different standards or publications. Based on this analysed knowledge, an ontological approach is developed with the idea to accommodate all the LOD systems. The development process is carried out based on the analysed requirements (CQ), which should be fulfilled by the adopted approach. Also to provide a clear understanding of the developed ontological approach for LOD representations, a demonstration example is elaborated based on the USA - BIMForum LOD system. The deliverable also explains the further framework capabilities corresponding to the BIM data management in terms of LOD-sensitive BIM data.

Furthermore, an analysis is carried on the requirements and needs to be addressed by the DICL ontology corresponding to renovation workflow representation and connection of LOD data to the renovation process. This analysis describes the involved components in renovation workflow concerning activities, tasks, and sequences, their interrelations to stakeholders involved, the required information, other resources, etc. To achieve this objective, the ontological approach is further extended based on the 6-dimensional representational schema including Activity, Stakeholder, BLS, LOD, BIM data and Use cases. This schema expresses the relationship between these six dimensions of the building renovation process.

Based on these investigations and the representational schema, a complete Digital Construction Lifecycle (DICL) ontology is developed and elaborated in the deliverable. The developed DICL ontology is also easily adaptable and applicable to the present BIM process since the representation of BIM data in different ontologies is within reach. The DICL ontology also brings flexibility, compliance and alignment capabilities through logical reasoning and knowledge inferencing in BIM data management. Moreover, the validation and evaluation of the developed DICL ontology were performed based on SPARQL queries. The results from these queries are presented either in the text or in complementing Annexes. With the defined and additional inherited knowledge capabilities, the developed ontology framework evidently presents its applicability and functionalities in terms of modelling BIM data in different LODs corresponding to renovation workflow.

Finally, the deliverable provides an overview of alignments and further dependencies with both the external and BIM4EEB ontologies. Furthermore, a demonstration example using a populated DICL ontology is presented in Annex V - Basic demonstration of LOD sensitive BIM data.

8 Publication bibliography

Abualdenien, J.; Borrmann, A. (2018): Multi-LOD model for describing uncertainty and checking requirements in different design stages. In Jan Karlshøj, Raimar Scherer (Eds.): eWork and eBusiness in Architecture, Engineering and Construction: CRC Press, pp. 187–195.

Abualdenien, Jimmy; Borrmann, André (2019): A meta-model approach for formal specification and consistent management of multi-LOD building models. In *Advanced Engineering Informatics* 40, pp. 135–153. DOI: 10.1016/j.aei.2019.04.003.

AIA Contract Documents (2013): G202-2013, Building Information Modeling Protocol. Available online at https://www.aiacontracts.org/contract-documents/19016-project-bim-protocol, updated on 1/5/2021, checked on 2/11/2021.

Allan, Luke; Menzel, Karsten (2009): Virtual Enterprises for Integrated Energy Service Provision. In Luis M. Camarinha-Matos, Hamideh Afsarmanesh, Iraklis Paraskakis (Eds.): Leveraging Knowledge for Innovation in Collaborative Networks. PRO-VE 2009, Thessaloniki, Greece, vol. 307. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg (IFIP Advances in Information and Communication Technology, 307), pp. 659–666.

Beetz, Jakob; van Leeuwen, Jos; Vries, Bauke de (2009): IfcOWL: A case of transforming EXPRESS schemas into ontologies. In *AIEDAM* 23 (1), pp. 89–101. DOI: 10.1017/S0890060409000122.

BIM FORUM (2020): LEVEL OF DEVELOPMENT (LOD) SPECIFICATION - PART 1 & COMMENTARY. For Building Models and Data. With assistance of Jim Bendrick, Will Ikerd, Jan Reinhardt. 2020th ed.: BIMForum. Available online at https://bimforum.org/lod/, checked on 11/2/2021.

BIM Wiki (2020): Level of detail for BIM. Design Buildings Wiki. Available online at https://www.designingbuildings.co.uk/wiki/Level_of_detail_for_BIM, updated on 9/30/2020, checked on 2/11/2021.

BIM4EEB (2021): BIM based fast toolkit for Efficient rEnovation in Buildings. D3.1 A BIM-based framework for building renovation using the linked data approach and ontologies. With assistance of VTT, Visualynk, TUD, PoliMi, Suite5, UCC, Rise: EU Commission.

Bonduel, Mathias (2018): Ontology for Property Management. Edited by Mads Holten Rasmussen, Maxime Lefrançois. W3C Community Group. Available online at https://w3c-lbd-cg.github.io/opm/, updated on 11/22/2018, checked on 4/20/2021.

UNI 11337-4, 2017: Building and civil engineering works – Digital management of building information process – Part 4: Evolution and development of information within models documents and objects.

ISO 23387:2020, 2020: Building information modelling (BIM) — Data templates for construction objects used in the life cycle of built assets — Concepts and principles.

ISO 23386:2020, 2020: Building information modelling and other digital processes used in construction — Methodology to describe, author and maintain properties in interconnected data dictionaries.

BS EN 17412-1:2020, 2020: Building Information Modelling. Level of Information Need. Part 1, Concepts and principles.

ISO 29481-1:2016, 2016: Building information models — Information delivery manual — Part 1: Methodology and format.

ISO 6707-1:2017, 2017: Buildings and civil engineering works — Vocabulary — Part 1: General terms.

Car, Nicholas J (2021): pyLODE - An OWL ontology documentation tool using Python and templating, based on LODE. Available online at https://pypi.org/project/pyLODE/, checked on 5/27/2021.

Chen, Weiwei; Chen, Keyu; Cheng, Jack C. P. (2018): Towards an Ontology-based Approach for Information Interoperability Between BIM and Facility Management. In Ian F. C. Smith, Bernd Domer (Eds.): Advanced computing strategies for engineering. 25th EG-ICE International Workshop 2018, Lausanne, Switzerland, vol. 10864. Cham, Switzerland: Springer (LNCS sublibrary: SL3 - Information systems and applications, incl. Internet/Web, and HCI, 10863-10864), pp. 447–469.

Di Biccari, Carla; Mangialardi, Giovanna; Lazoi, Mariangela; Corallo, Angelo (2018): Configuration Views from PLM to Building Lifecycle Management. In Paolo Chiabert (Ed.): Product lifecycle management to support industry 4.0. 15th IFIP WG 5.1 International Conference, PLM 2018, Turin, Italy, July 2-4, 2018 : proceedings, vol. 540. Cham: Springer (IFIP Advances in Information and Communication Technology, 540), pp. 69–79.

Dong, B; Lam, K P; Huang, Y C; Dobbs, G M (2007): A comparative study of the IFC and gbXML informational infrastructures for data exchange in computational design support environments. In *Building Simulation*, pp. 1530–1537.

BS EN 16310:2013, 2013: Engineering services — Terminology to describe engineering services for buildings, infrastructure and industrial facilities.

Garijo, Daniel; Poveda-Villalón, María (2020): Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web. Available online at http://arxiv.org/pdf/2003.13084v1.

Guzmán, Elizabeth; Zhu, Zhenhua (2014): Interoperability between Building Design and Building Energy Analysis. In Raymond Issa Issa, Ian Flood (Eds.): Computing in Civil and Building Engineering (2014). 2014 International Conference on Computing in Civil and Building Engineering. Orlando, Florida, United States, June 23-25, 2014. [Place of publication not identified]: American Society of Civil Engineers, pp. 17–24.

HOAI, 2001: Honorarordnung für Architekten und Ingenieure.

Hooper, Martin (2015): Automated model progression scheduling using level of development. In *Construction Innovation* 15 (4), pp. 428–448. DOI: 10.1108/CI-09-2014-0048.

ISO 16739-1:2018, 2018: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema. Available online at https://www.iso.org/standard/70303.html, checked on 2/24/2021.

Karlapudi, Janakiram; Menzel, Karsten (2020): Analysis on automatic generation of BEPS models from BIM model. In Michael Monsberger, Christina Johanna Hopfe, Markus Krüger, Alexander Passer (Eds.): BauSIM 2020 - 8th Conference of IBPSA Germany and Austria. 2020th ed. Austria: Verlag der Technischen Universität Graz, pp. 535–542. Available online at https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-735477, checked on 2/24/2021.

Karlapudi, Janakiram; Menzel, Karsten; Törmä, Seppo; Hryshchenko, Andriy; Valluru, Prathap (2020): Enhancement of BIM Data Representation in Product-Process Modelling for Building Renovation. In Felix Nyffenegger, José Ríos, Louis Rivest, Abdelaziz Bouras (Eds.): Product Lifecycle Management Enabling Smart. 17th ifip wg 5.1, vol. 594. [S.l.]: Springer (IFIP Advances in Information and Communication Technology), pp. 738–752.

Karlapudi, Janakiram; Shetty, Suraj (2019): A methodology to determine and classify data sharing requirements between OpenBIM models and energy simulation models. In Maximilian Sternal, Lucian-

Constantin Ungureanu, Laura Böger, Christoph Bindal-Gutsche (Eds.): 31. Forum Bauinformatik. With assistance of Maximilian Sternal, Lucian-Constantin Ungureanu, Laura Böger, Christoph Bindal-Gutsche. Berlin: Universitätsverlag der TU Berlin, pp. 331–338. Available online at https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-735487, checked on 2/24/2021.

Keller, Martin; O'Donnel, James; Menzel, Karsten; Keane, Marcus (2008): Integrating the Specification, Acquisition and Processing of Building Performance Information. In *Tinshhua Sci. Technol.* 13 (1), pp. 1–6.

BIMFORUM, 2020: Level of Development (LOD) Specification Part I & Commentary - For Building Information Models and Data.

Li, Xiao; Wu, Peng; Shen, Geoffrey Qiping; Wang, Xiangyu; Teng, Yue (2017): Mapping the knowledge domains of Building Information Modeling: A bibliometric approach. In *Automation in Construction* 84, pp. 195–206. DOI: 10.1016/j.autcon.2017.09.011.

Lilis, Georgios N.; Giannakis, Georgios I.; Rovas, Dimitrios V. (2017): Automatic generation of second-level space boundary topology from IFC geometry inputs. In *Automation in Construction* 76, pp. 108–124. DOI: 10.1016/j.autcon.2016.08.044.

Mangialardi, Giovanna; Di Biccari, Carla; Pascarelli, Claudio; Lazoi, Mariangela; Corallo, Angelo (2017): BIM and PLM Associations in Current Literature. In José Ríos, Alain Bernard, Abdelaziz Bouras, Sebti Foufou (Eds.): Product lifecycle management and industry of the future. 14th IFIP WG 5.1 international conference, PLM 2017, Seville, Spain, July 10-12, 2017, vol. 517. Cham: Springer (IFIP Advances in Information and Communication Technology, 517), pp. 345–357.

ISO 22263:2008, 2008: Organization of information about construction works — Framework for management of project information.

Papadonikolaki, E.; Leon, M.; Mahamadu, A. M. (2018): BIM solutions for construction lifecycle: A myth or a tangible future? In Jan Karlshøj, Raimar Scherer (Eds.): eWork and eBusiness in Architecture, Engineering and Construction: CRC Press, pp. 321–328.

Pauwels, Pieter; Zhang, Sijie; Lee, Yong-Cheol (2017): Semantic web technologies in AEC industry: A literature overview. In *Automation in Construction* 73, pp. 145–165. DOI: 10.1016/j.autcon.2016.10.003.

Ramaji, Issa J.; Memari, Ali M. (2016): Interpreted Information Exchange: Systematic Approach for BIM to Engineering Analysis Information Transformations. In *J. Comput. Civ. Eng.* 30 (6), p. 4016028. DOI: 10.1061/(ASCE)CP.1943-5487.0000591.

RIBA 2020, 2020: RIBA Plan of Work 2020 Overview. Available online at http://www.ribaplanofwork.com/, checked on 4/14/2021.

Saaksvuori, Antti; Immonen, Anselmi (2004): Product Lifecycle Management. Berlin, Heidelberg: Springer Berlin Heidelberg; Imprint; Springer.

PAS 1192-2:2013, February 2013: Specification for information management for the capital/delivery phase of construction projects using building information modelling.

PAS 1192-2, 2013: Specification for information management for the capital/delivery phase of construction pro-jects using building information modelling.

PAS 1192-3, 2014: Specification for information management for the operational phase of assets using building information modelling.

PAS 1192-3:2014, March 2014: Specification for information management for the operational phase of assets using building information modelling.

Treldal, Niels; Vestergaard, F; Karlshoj, J (2016): Pragmatic Use of LOD – a Modular Approach. In Symeon Christodoulou, Raimar Scherer (Eds.): EWork and eBusiness in Architecture, Engineering and Construction. Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2016), Limassol, Cyprus, 7-9 September 2016, vol. 2016. London: CRC Press, pp. 129–136. Available online at https://backend.orbit.dtu.dk/ws/files/127947941/ecppm2016_paper_54.pdf, checked on 2/24/2021.

Venugopal, M.; Eastman, C. M.; Sacks, R.; Teizer, J. (2012): Semantics of model views for information exchanges using the industry foundation class schema. In *Advanced Engineering Informatics* 26 (2), pp. 411–428. DOI: 10.1016/j.aei.2012.01.005.

Xu, Zhao; Abualdenien, Jimmy; Liu, Hao; Kang, Rui (2020): An IDM-Based Approach for Information Requirement in Prefabricated Construction. In *Advances in Civil Engineering* 2020, pp. 1–21. DOI: 10.1155/2020/8946530.

Zadeh, Puyan A.; Wang, Guan; Cavka, Hasan B.; Staub-French, Sheryl; Pottinger, Rachel (2017): Information Quality Assessment for Facility Management. In *Advanced Engineering Informatics* 33, pp. 181–205. DOI: 10.1016/j.aei.2017.06.003.

9 Annex I – LOD framework validation

Table 14: LOD - Queries and their results – Part 2

CQ3:- What is the relation of LOD Classification-Levels with other levels of the USA LOD scale?

Query	?System	?Sub_type	?Scale
SELECT ?System ?Sub_type ?Scale Where{ ?System dicl:hasLevel ?Sub_type. FILTER(?System =diclvl:USA_BIMForum). ?Sub_type dicl:hasSubLevel ?Scale . }ORDER BY ASC(?Scale)	:USA_BIMForum	:AsDesigned	:LOD_100
	:USA_BIMForum	:AsDesigned	:LOD_200
	:USA_BIMForum	:AsDesigned	:LOD_300
	:USA_BIMForum	:AsDesigned	:LOD_350
	:USA_BIMForum	:AsDesigned	:LOD_400
	:USA_BIMForum	:AsBuilt	:LOD_500

CQ4: What is the relation between LOD scales?
CQ4.1:- What are the next levels for each level in the USA LOD system?

Query	?system	?level	?Nextlevels
SELECT ?system ?level ?Nextlevels WHERE { ?system dicl:hasLevel ?level. optional{?level dicl:hasNextLevel ?Nextlevels.} Filter(?system =diclvl:USA_BIMForum) }ORDER BY ASC(?level)	:USA_BIMForum	:AsBuilt	
	:USA_BIMForum	:AsDesigned	:AsBuilt
	:USA_BIMForum	:LOD_100	:LOD_500
	:USA_BIMForum	:LOD_100	:LOD_300
	:USA_BIMForum	:LOD_100	:LOD_400
	:USA_BIMForum	:LOD_100	:LOD_200
	:USA_BIMForum	:LOD_100	:LOD_350
	:USA_BIMForum	:LOD_200	:LOD_500
	:USA_BIMForum	:LOD_200	:LOD_300
	:USA_BIMForum	:LOD_200	:LOD_400
	:USA_BIMForum	:LOD_200	:LOD_350
	:USA_BIMForum	:LOD_300	:LOD_500
	:USA_BIMForum	:LOD_300	:LOD_400
	:USA_BIMForum	:LOD_300	:LOD_350
	:USA_BIMForum	:LOD_350	:LOD_500
	:USA_BIMForum	:LOD_350	:LOD_400
	:USA_BIMForum	:LOD_400	:LOD_500
	:USA_BIMForum	:LOD_500	

CQ4: What is the relation between LOD scales?
CQ4.2:- What are the previous levels for each level in the USA LOD system?

```
SELECT ?system ?level ?Previouslevels
WHERE {
?system dicl:hasLevel
?level.
optional{?level
dicl:hasPreviousLevel
?Previouslevels.}
Filter(?system =
:USA_BIMForum)
}ORDER BY ASC(?level)
```

?system	?level	?Previouslevels
:USA_BIMForum	:AsBuilt	:AsDesigned
:USA_BIMForum	:AsDesigned	
:USA_BIMForum	:LOD_100	
:USA_BIMForum	:LOD_200	:LOD_100
:USA_BIMForum	:LOD_300	:LOD_100
:USA_BIMForum	:LOD_300	:LOD_200
:USA_BIMForum	:LOD_350	:LOD_300
:USA_BIMForum	:LOD_350	:LOD_100
:USA_BIMForum	:LOD_350	:LOD_200
:USA_BIMForum	:LOD_400	:LOD_300
:USA_BIMForum	:LOD_400	:LOD_100
:USA_BIMForum	:LOD_400	:LOD_200
:USA_BIMForum	:LOD_400	:LOD_350
:USA_BIMForum	:LOD_500	:LOD_300
:USA_BIMForum	:LOD_500	:LOD_400
:USA_BIMForum	:LOD_500	:LOD_100
:USA_BIMForum	:LOD_500	:LOD_200
:USA_BIMForum	:LOD_500	:LOD_350

Query:- What are different LOD systems and their respective levels?

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

	System	Sub_type	Scale
1	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_A
2	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_B
3	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_C
4	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_D
5	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_E
6	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_F
7	diclvl:Italy_Level_of_Development	diclvl:LOG_Geometrical_Object	diclvl:LOG_G

8	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_A
9	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_B
10	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_C
11	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_D
12	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_E
13	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_F
14	diclvl:Italy_Level_of_Development	diclvl:LOI_Information_Object	diclvl:LOI_G
15	diclvl:UK_Level_of_Definition	diclvl:Level_of_Detail	diclvl:LOD_1
16	diclvl:UK_Level_of_Definition	diclvl:Level_of_Detail	diclvl:LOD_2
17	diclvl:UK_Level_of_Definition	diclvl:Level_of_Detail	diclvl:LOD_3
18	diclvl:UK_Level_of_Definition	diclvl:Level_of_Detail	diclvl:LOD_4
19	diclvl:UK_Level_of_Definition	diclvl:Level_of_Detail	diclvl:LOD_5
20	diclvl:UK_Level_of_Definition	diclvl:Level_of_Information	diclvl:LOI_1
21	diclvl:UK_Level_of_Definition	diclvl:Level_of_Information	diclvl:LOI_2
22	diclvl:UK_Level_of_Definition	diclvl:Level_of_Information	diclvl:LOI_3
23	diclvl:UK_Level_of_Definition	diclvl:Level_of_Information	diclvl:LOI_4
24	diclvl:UK_Level_of_Definition	diclvl:Level_of_Information	diclvl:LOI_5
25	diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_100
26	diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_200
27	diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_300
28	diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_350
29	diclvl:USA_BIMForum	diclvl:AsDesigned	diclvl:LOD_400
30	diclvl:USA_BIMForum	diclvl:AsBuilt	diclvl:LOD_500

Here some images are provided related to the example queries above. The objective of these images to compare the query results before and after the inclusive of inferred data. Only some queries from above table is considered this purpose.

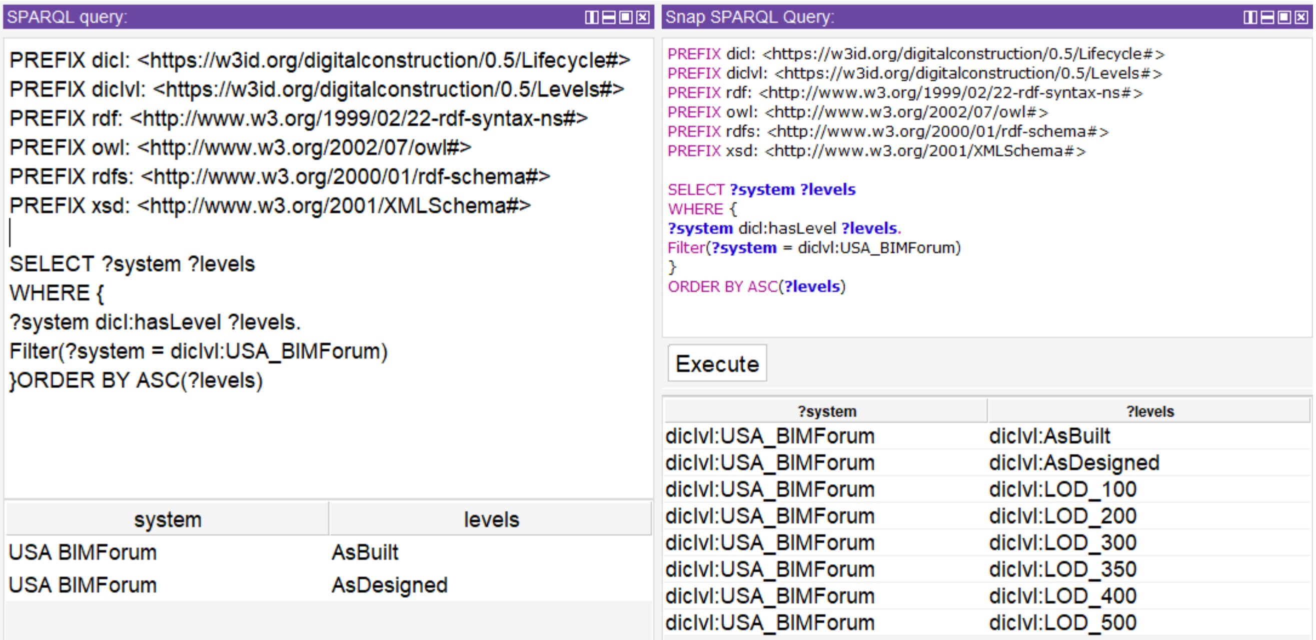


Figure 18: LOD levels according to the USA BIMForum System

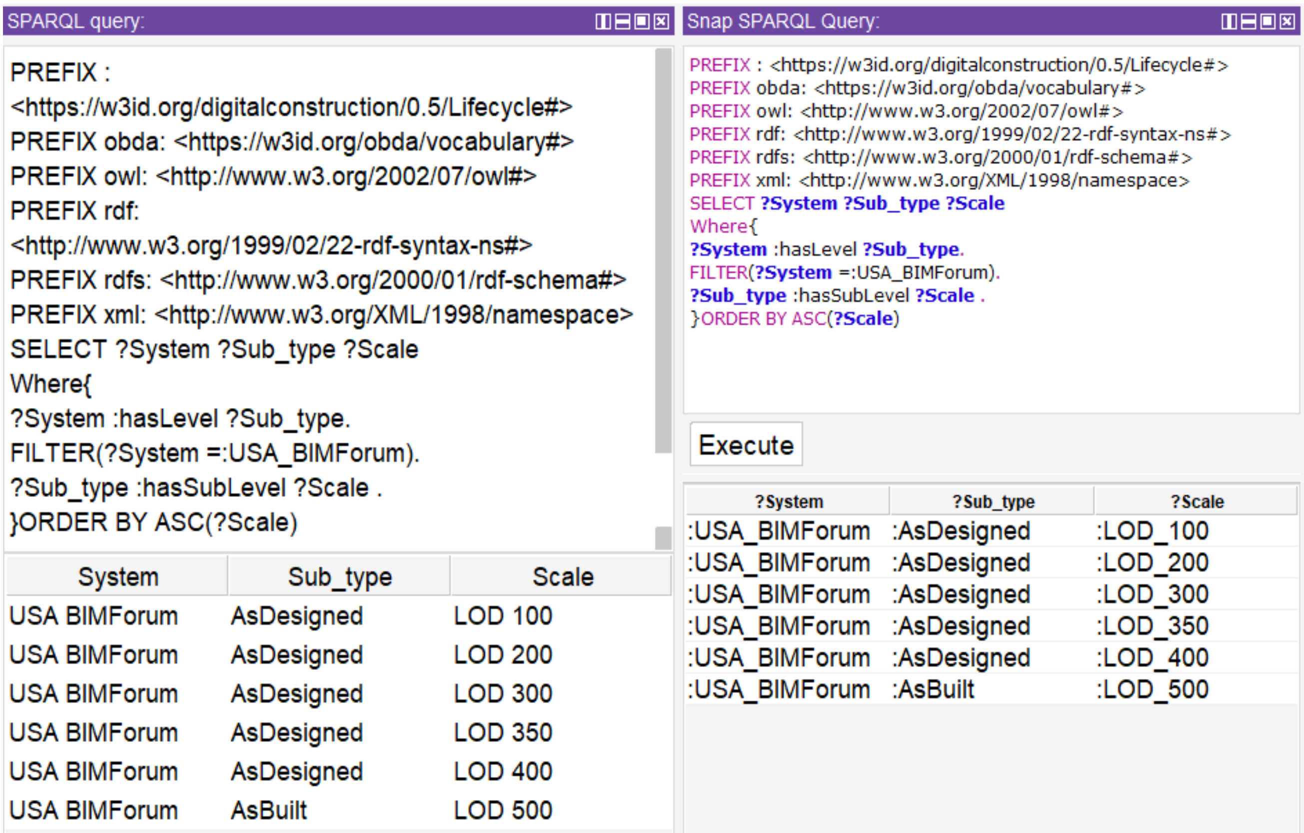


Figure 19: Relation of LOD Classification-Levels with other levels of the USA LOD scale

SPARQL query:

```
PREFIX dicl:
<https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX obda: <https://w3id.org/obda/vocabulary#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>

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

system	level	Nextlevels
USA BIMForum	AsBuilt	
USA BIMForum	AsDesigned	AsBuilt

Snap SPARQL Query:

```
PREFIX dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX obda: <https://w3id.org/obda/vocabulary#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>

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

Execute

?system	?level	?Nextlevels
dicl:USA_BIMForum	dicl:AsBuilt	
dicl:USA_BIMForum	dicl:AsDesigned	dicl:AsBuilt
dicl:USA_BIMForum	dicl:LOD_100	dicl:LOD_500
dicl:USA_BIMForum	dicl:LOD_100	dicl:LOD_300
dicl:USA_BIMForum	dicl:LOD_100	dicl:LOD_400
dicl:USA_BIMForum	dicl:LOD_100	dicl:LOD_200
dicl:USA_BIMForum	dicl:LOD_100	dicl:LOD_350
dicl:USA_BIMForum	dicl:LOD_200	dicl:LOD_500
dicl:USA_BIMForum	dicl:LOD_200	dicl:LOD_300
dicl:USA_BIMForum	dicl:LOD_200	dicl:LOD_400
dicl:USA_BIMForum	dicl:LOD_200	dicl:LOD_350
dicl:USA_BIMForum	dicl:LOD_300	dicl:LOD_500
dicl:USA_BIMForum	dicl:LOD_300	dicl:LOD_400
dicl:USA_BIMForum	dicl:LOD_300	dicl:LOD_350
dicl:USA_BIMForum	dicl:LOD_350	dicl:LOD_500
dicl:USA_BIMForum	dicl:LOD_350	dicl:LOD_400
dicl:USA_BIMForum	dicl:LOD_400	dicl:LOD_500
dicl:USA_BIMForum	dicl:LOD_500	

Figure 20: Next levels for each level in the USA LOD system

SPARQL query:

```
PREFIX dicl:
<https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX obda: <https://w3id.org/obda/vocabulary#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>

SELECT ?system ?level ?Previouslevels
WHERE {
?system dicl:hasLevel ?level.
optional{?level dicl:hasPreviousLevel ?Previouslevels.}
Filter(?system = dicl:USA_BIMForum)
}ORDER BY ASC(?level)
```

system	level	Previouslevels
USA BIMForum	AsBuilt	
USA BIMForum	AsDesigned	

Snap SPARQL Query:

```
PREFIX dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX obda: <https://w3id.org/obda/vocabulary#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>

SELECT ?system ?level ?Previouslevels
WHERE {
?system dicl:hasLevel ?level.
optional{?level dicl:hasPreviousLevel ?Previouslevels.}
Filter(?system = dicl:USA_BIMForum)
}ORDER BY ASC(?level)
```

Execute

?system	?level	?Previouslevels
dicl:USA_BIMForum	dicl:AsBuilt	dicl:AsDesigned
dicl:USA_BIMForum	dicl:AsDesigned	
dicl:USA_BIMForum	dicl:LOD_100	
dicl:USA_BIMForum	dicl:LOD_200	dicl:LOD_100
dicl:USA_BIMForum	dicl:LOD_300	dicl:LOD_100
dicl:USA_BIMForum	dicl:LOD_300	dicl:LOD_200
dicl:USA_BIMForum	dicl:LOD_350	dicl:LOD_300
dicl:USA_BIMForum	dicl:LOD_350	dicl:LOD_100
dicl:USA_BIMForum	dicl:LOD_350	dicl:LOD_200
dicl:USA_BIMForum	dicl:LOD_400	dicl:LOD_300
dicl:USA_BIMForum	dicl:LOD_400	dicl:LOD_100
dicl:USA_BIMForum	dicl:LOD_400	dicl:LOD_200
dicl:USA_BIMForum	dicl:LOD_400	dicl:LOD_350
dicl:USA_BIMForum	dicl:LOD_500	dicl:LOD_300
dicl:USA_BIMForum	dicl:LOD_500	dicl:LOD_400
dicl:USA_BIMForum	dicl:LOD_500	dicl:LOD_100
dicl:USA_BIMForum	dicl:LOD_500	dicl:LOD_200
dicl:USA_BIMForum	dicl:LOD_500	dicl:LOD_350

Figure 21: Previous levels for each level in the USA LOD system

10 Annex II – BLS framework validation

SPARQL query:

```
PREFIX dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?Framework ?Stage ?NextStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Framework=dicl:BS_EN_16310) .
Filter(?Stage=dicl:4.BS_EN_Construction) .
?Stage dicl:hasNextStage ?NextStages .
}
ORDER BY ASC(?NextStages)
```

Framework	Stage	NextStages
BS_EN_16310	4.BS_EN_Construction	5.BS_EN_Use

Snap SPARQL Query:

```
PREFIX dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?Framework ?Stage ?NextStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Framework=dicl:BS_EN_16310) .
Filter(?Stage=dicl:4.BS_EN_Construction) .
?Stage dicl:hasNextStage ?NextStages .
}
ORDER BY ASC(?NextStages)
```

Execute

?Framework	?Stage	?NextStages
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:5.BS_EN_Use
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:5.1.BS_EN_Operation
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:5.2.BS_EN_Maintenance
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:6.BS_EN_End_of_Life
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:6.1.BS_EN_Revamping
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:6.2.BS_EN_Dismantling

Figure 22: The next stages of the stage "Construction" according to BS EN 16310

SPARQL query:

```
PREFIX dicl:
<https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?Framework ?Stage ?PreviousStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Framework=dicl:BS_EN_16310) .
Filter(?Stage=dicl:4.BS_EN_Construction) .
?Stage dicl:hasPreviousStage ?PreviousStages .
}
ORDER BY ASC(?PreviousStages)
```

Framework	Stage	PreviousStages

Snap SPARQL Query:

```
PREFIX dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?Framework ?Stage ?PreviousStages
WHERE {
?Framework dicl:hasStage ?Stage .
Filter(?Framework=dicl:BS_EN_16310) .
Filter(?Stage=dicl:4.BS_EN_Construction) .
?Stage dicl:hasPreviousStage ?PreviousStages .
}
ORDER BY ASC(?PreviousStages)
```

Execute

?Framework	?Stage	?PreviousStages
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:0.BS_EN_Initiative
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:0.1.BS_EN_Market_study
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:0.2.BS_EN_Business_case
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:1.BS_EN_Initiation
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:1.1.BS_EN_Project_initiation
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:1.2.BS_EN_Feasibility_study
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:1.3.BS_EN_Project_definition
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:2.BS_EN_Design
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:2.1.BS_EN_Conceptual_design
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:2.2.BS_EN_Preliminary_design
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:2.3.BS_EN_Developed_design
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:2.4.BS_EN_Technical_design
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:2.5.BS_EN_Detailed_design
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:3.BS_EN_Procurement
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:3.1.BS_EN_Procurement
dicl:BS_EN_16310	dicl:4.BS_EN_Construction	dicl:3.2.BS_EN_Construction_contracting

Figure 23: Previous stages of the stage "Construction" according to BS EN 16310

11 Annex III – Integrated Product-Process modelling examples

The tables below present more comprehensive examples of the Entities-BLS-matrix.

Use Cases as dynamic combination of specific Activities	Use Cases Names	Abbreviation	Activities supporting Use Cases 1 - 3												Use Cases Definition
	Use Case 1	UC1	3, 4	13, 14, 16, 20, 21, 32, 47, 48	55, 56, 60, 61	73, 74, 75	95, 96		118			-	-	178, 184, 185	Data Acquisition for HVAC Design, Operation and Efficiency Management
	Use Case 2	UC2	-	17, 18, 19, 27	63-68	76-79, 83, 84, 86-91	97, 98, 99, 100, 101, 102, 103, 104, 105, 107-110		119, 120, 121, 122–129, 131, 132, 134, 135			141, 142, 160, 161	-	181, 182, 183, 187, 191, 195	HVAC Design, Operation and Efficiency Mgmt. for Fast Track Renovation Operation
	Use Case 3	UC3	-	22, 28, (29, 30), 33, 34, 35, 37, 40, 44			111, 112, 116, 117					143, 144, 145, 146, 147, 148, 155, 156, 159	166, 167, 168, 169, 170, 173, 174, 175	189, 192	Fast Track Renovation Operation

BIM Entities, Attributes, P_sets			For Scenarios № see Tables 2-4, 6-8 in D3.3 and Tables 4-12 in D2.1																									
			Building Lifecycle Stages (BS EN 16310)																				LOD					
		BLS №	0		1			2					3		4					5		6						
		BLS Name	Initiative		Initiation			Design					Procurement		Construction					Use		End of Life						
Object №	Entities / Sets	Attributes and related Psets	Market study	Business case	Project initiation	Feasibility study	Project definition	Concept Design	Preliminary Design	Developed Design	Technical Design	Detailed Design	Procurement	Construction Contracting	Pre-construction	Construction	Commissioning	Handover	Regulatory Approval	Operation	Maintenance	Revamping	Dismantling	LOD 100	LOD 200	LOD 300	LOD 350	LOD 400	LOD 500
Building Elements																													
1	External Wall	IfcWall	LOD100	LOD200	LOD100	LOD300	LOD350	LOD100	LOD300	LOD350	LOD400	LOD400	LOD300	LOD350	LOD400	LOD300	LOD500	LOD300	LOD350	LOD300	LOD400	LOD500	LOD400	3	3	2, 3	2, 3	1, 2	1
2	Material	IfcMaterial	LOD300	LOD100	LOD100	LOD300	LOD300	LOD100	LOD300	LOD350	LOD300	LOD400	LOD300			LOD300					LOD300		LOD300						
3	Internal Wall	IfcWall	LOD200		LOD100		LOD300	LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						
4	Window	IfcWindow	LOD300		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						
5	Door	IfcDoor	LOD200		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						
6	Roof	IfcRoof	LOD200		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						
7	Slab	IfcSlab	LOD200		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300	1, 2, 3					
8	Curtain Wall	IfcCurtainWall	LOD200		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						
9	Shading	IfcShadingDevice	LOD100		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						
10	IfcBeam (Archiectural Point	IfcBeam	LOD200		LOD100			LOD100	LOD300	LOD350		LOD400				LOD300					LOD300		LOD300						

Figure 24: BIM entities – LOD matrix example

The table below provides an instrument to map stakehaolders involvement (rows) in Building Lifecycle Stages (columns) by iedntifying dedicatzed activities (cells).

Stakeholders and their involvement in Activities (defined in BIM4EEB D2.1)																						
№	Stakeholders / BLS	Building Lifecycle Stages (BS EN 16310)																				
		0		1			2					3		4					5		6	
		0.1	0.2	1.1	1.2	1.3	2.1	2.2	2.3	2.4	2.5	3.1	3.2	4.1	4.2	4.3	4.4	4.5	5.1	5.2	6.1	6.2
1	Inhabitant / End-user	1, 2		41		40						42	43	141, 142		154, 155, 156			168, 169, 170			
2	Client / Owner	3, 4, 5, 7	9	10, 11	15	52, 53													170, 171			182
3	Client' Adviser																					
4	Project Leader																					
5	Site Surveyor																					
6	Cost Consultant / QS				36																	
7	Health and Safety Adviser																				180	
8	Information Manager																					
9	Contract Administrator													139, 140								
10	Tenderer					62																
11	Lead Designer				35		54, 67, 68	90, 91	111, 112		128, 129											
12	Architectural Designer																					
13	Structural Designer						160	160	160	161	161					160, 161						
14	Building Services Designer						160	160	160	161	161											
15	Sustainability Adviser																		167	167		193
16	Facilities Mgmt. Advisor																		166	172		
17	Technical Adviser						178	178				178		178		178					178	176
18	Contractor																					194, 195
19	Local Authority						74	74	117													188
20	Bank or third party financier		50	50	50	51																
21	Work supervisor																					
22	Tester (Commissioner)																					
23	Fire safety designer																					
24	Lanscape designer																					
25	Accoustic Consultant																					
26	Cladding Specialist																					
27	Interior Designer																					
28	Lighting Designer																					
29	Security Adviser																					
30	Access Consultant																					
31	Supplier																					
32	Master Planner																					
33	Planning Consultant																					
34	Party wall surveyor																					
35	Sub-Contractor																					
36	Maintenance Planner																					
37	Construction Leader																					
38	Operational Leader																					

Figure 25: Stakeholders and their Activities w.r.t. BLS

The table below 'formalises' the so-called "Swim Lane Approach" by specifying the role (columns) of stakeholders (cell) in different activities. This is the densest level of granularity that can be provided in terms of the A-Box population.

MACHINE READABLE "SWIM LANE" APPROACH			
Activity №	Stakeholder's №		
	Information Provider	Information Processor	Information Consumer
1	1	2	16
2	1	2	2
3	2	2	2
4	2	2	2
5	2	3	3
6	3	2	2
7	2	2	3
8	3	2	2
9	2	2	2
10	2	2	2
11	2	2	4
12	4	4	1, 2
13	4	4	4
14	4	4	2
15	2	5	5
16	5	5	4
17	5	5	4
18	5	5	4
19	5	5	4
20	5	5	4
21	5	4	4
22	4	2	2
23	4	4	2
24	4	2	2
25	4	2	2
26	4	4	2
27	4	4	2
28	4	2, 4	2, 4
29	4	2, 4	2, 4
30	4		

Figure 26: Stakeholders and their Activities w.r.t. information use

12 Annex IV – Additional Use Case Definitions

Use Case 2: HVAC Design, Operation & Efficiency Management for Fast Track Renovation Operation

The list of activities involved in this use case describes the present condition of the systems installed in a building. These activities are also involved in systems' operation and efficiency management in terms of creating execution plans, strategies and systems' enhancement techniques. These activities are also covering tendering, construction and buildings' operation techniques.

Table 15: Stakeholders and their involvement in Use Case 2

BLS / Stakeholders	0	1	2.1	2.2	2.3	2.5	4	5	6
Project Leader		27	63 - 66	83, 84, 86 to 89	97, 98, 101-105, 107-110	121, 122 - 126, 127, 131, 132, 134, 135	142		181, 183, 187, 191
Lead Designer			67, 68	90, 91		128, 129			
Architectural Designer				76, 77, 78, 79	99, 100	119, 120, 123, 124			
Structural Designer							160, 161		
Building Services Designer							160, 161		
Site Surveyor		17, 18, 19							
HVAC-Engineer							160, 161		
Electrical Engineer							160, 161		
Contractor									195
Client									182
Owner									182
Inhabitant							141		
End-user							141		

Use Case 3: Fast Track Renovation Operation

The use case consists of those activities allowing to speed up the renovation process by setting up some strategies, basic planning principles and efficient investigation of possibilities. The activities related to cost-efficient construction planning and schedules. Those activities diverted on awareness of the renovation status are also included in this Use Case. Inspection and monitoring of day-to-day activities in the construction (progress) and checking of their compliance with plans are also the major activities that play an important role in the efficient and fast renovation process.

Table 16: Stakeholders and their involvement in Use Case 3

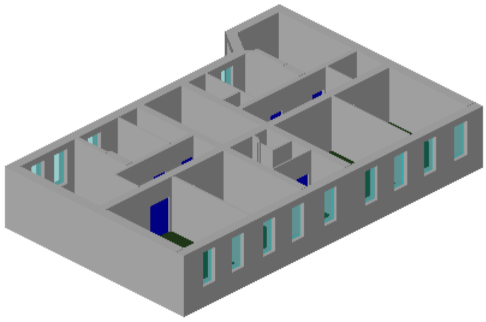
BLS / Stakeholders	0	1	2.1	2.2	2.3	2.5	4	5	6
Project Leader		22, 28-30, 33, 34, 37			116		143-148, 159	174, 175	189, 192
Lead Designer		35			111, 112				
FM Advisor								166, 173	
Sustainable Adviser								167	
Owner		44						170	
Inhabitant		40					155, 156	168-170	
End-user		40					155, 156	168, 169	
Local Authority					117				

13 Annex V - Basic demonstration of LOD sensitive BIM data

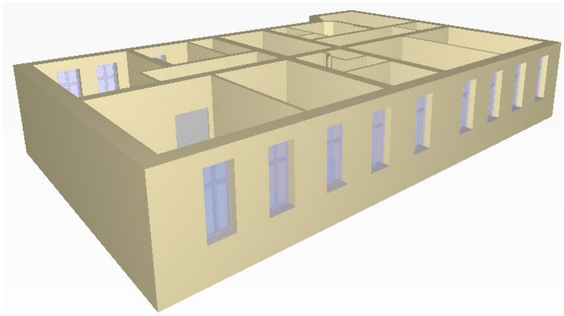
This annexe demonstrates the evolution of BIM data according to different LOD. The demonstration explores the storage and filtering of LOD based BIM data using the Digital Construction Lifecycle ontology.

For this demonstration, sample BIM models from the Polish demo site are considered. These files were generated by WP8 partners (Prochem) for this specific demonstration. The model files are

1. Polish_site_2nd_floor_LOD_200_fix.ifc;
2. Polish_site_2nd_floor_LOD_300_fix.ifc.



Polish_site_2nd_floor_LOD_200_fix.ifc.



Polish_site_2nd_floor_LOD_300_fix.ifc.

Figure 27: Demo BIM Model

Each example file contains the information regarding the Polish demo site's 2nd floor concerning the specified LOD level. Initially, an analysis of the generated BIM models was performed. The analysis results are captured in Table 17 and Table 18. The first table intends to specify the number of instances within the BIM models with different LOD levels. The observations from Table 17 clarify the only changes in the material or layer information assigned to the building objects by using IfcRelAssociatesMaterial.

Table 17: Comparison of BIM Model Instances

BIM Objects	Amount of Instances	
	LOD 200 Model	LOD 300 Model
Entities	140	140
• IfcBuilding	1	1
• IfcBuildingStorey	1	1
• IfcDoor	17	17
• IfcOpeningElement	35	35
• IfcProject	1	1
• IfcSite	1	1
• IfcSlab[Floor]	1	1
• IfcSpace	34	34
• IfcWall	30	30
• IfcWindow	16	16

BIM Objects	Amount of Instances	
	LOD 200 Model	LOD 300 Model
• IfcZone	3	3
Relations	628	640
• IfcRelAggregates	4	4
• IfcRelAssignsToGroup	3	3
• IfcRelAssociatesClassification	2	1
• IfcRelAssociatesMaterial	76	87
• IfcRelConnectsPathElements	47	47
• IfcRelContainedInSpatialStructure	1	1
• IfcRelDefinesByProperties	372	372
• IfcRelDefinesByType	55	57
• IfcRelFillsElement	33	33
• IfcRelVoidsElement	35	35
EntityTypes	5	5
• IfcDoorStyle	13	13
• IfcSlabType	1	1
• IfcSpaceType	34	34
• IfcWallType	4	4
• IfcWindowStyle	3	5

In later stages, for more understanding of the incremental changes in both models, the observations are extended to the level of attributes. For this, a single instance from each entity (mentioned in Table 17) is considered and the results from these observations are illustrated in Table 18.

This table represents the gradual increment within the object properties (attributes) along with the increase of Levels of detail. It also observed the growth of the object's construction details (material layers or materials). From this basic analysis of these BIM models, the following scenarios are observed.

- Scenario 1:- there is no change in the BIM Objects or entities (wall, slab, etc.) but the object property growth is observed along with the increment of LOD levels;
- Scenario 2:- Observed a growth in BIM object's construction details in terms of Layers and Material information

Also, for the development of Digital Construction Lifecycle ontology, these scenarios are considered as development requirements. The following ontology demonstration also provides clear understandings of the developed ontology by addressing the above-mentioned scenarios.

Table 18: Comparison of BIM Model attributes

Parameters	**LOD 200 Model**	**LOD 300 Model**
Wall.1.1 – Basic Wall:500 + 100: 2145690		
• Height	3.8 m	3.9 m
• Length	22.85 m	23.13 m
• Thickness	600 mm	640 mm
• Layers or Material	Brick wall, Thermal isolation	Cement plaster, Strofoam, Brick, Plaster
• Thermal transmittance	0.25	0.26
• FireRating	-	REI60
• Description	Exterior Wall	Exterior Wall – Brick 500
• Model	-	Type 1
Space 1.1 - Pantry		
• Height	3.00 m	3.10 m
• Length	2.57 m	2.58 m
• Width	1.32 m	1.28 m
• Perimeter	7.80 m	7.73 m
• Volume	10.22 cu.m	10.26 cu.m
• Area	3.41 sq.m	3.31 sq.m
• Comments	-	Second Floor Spaces
Slab 1.1 – Floor:400:2271004		
• Area	262,21 sq.m	266,83 sq.m
• Gross Area	262,21 sq.m	266,83 sq.m
• Volume	104,88 cu.m	106,73 cu.m
• Thickness	350 mm	400 mm
• Layers or Material	Floor,Structure	Terracotta, Screed, Stryrofoam, Concrete
• FireRating	-	REI60
• Description	Iner-story Slab	Iner-story Slab 400
• Model	-	Type 1
Window 1.1 - Window 7:100 x 220: 2145699		
• Height	2.04 m	2.2 m
• Width	0.95 m	1 m
• Area	2.0 sq.m	2.2 sq.m
• Layers or Material	Generic	Glass, PCV
• Description	Exterior window	Exterior window 2200x1000
• Model	-	Type 1
Door 1.1 - Doors_single:800 x 200: 2342950		
• Height	2.05 m	2.1 m
• Width	900 mm	950 mm
• Area	1,84 sq.m	1,99 sq.m
• Layers or Material	Generic	Wood, Aluminium
• FireRating	-	EI15
• Description	Interior Door	Interior Door 900x2000
• Model	-	Type 1

In general, the demonstration workflow is majorly divided into three different parts. Such is

1. Data Conversion;
2. Data Population; and
3. Data Validation

In the Data Conversion process, the received IFC files are converted into respective OWL files using the BIMMS platform. This is an automated conversion process established in the BIMMS platform. So this conversion process is not described within this deliverable.

The next part of the demonstration describes the population of data from the BIM models (OWL files) to DICL ontology.

The last part explores the filtering of data using LOD levels. This demonstration work has been carried out with the help of the SPARQL queries. Detailed discussions concerning the workflow along with the query profiles are provided in the subsequent sections. Due to the high amount of data, the documentation of the demonstration is restricted to Wall Objects (wall entities) but the workflow is valid for the entire BIM model.

13.1.1 Populating the DICL ontology

The population of already available information from BIM models to the required BIM4EEB ontologies is the major task in the complete demonstration process. In this demonstration process, a set of iterative SPARQL profiles are used for populating the data but for the documentation the final successful and amalgamated profiles are presented. In the process of data population, at first SPARQL SELECT queries are used to verify the available data in BIM models and then extended them into SPARQL CONSTRUCT and SPARQL INSERT queries to construct the extracted data into Digital Construction Lifecycle ontology.

Table 19: The query for the extraction of wall object data

1) Extraction of wall entities	2) Extraction of Object properties

?WallInstance rdf:type ifcowl:IfcWall .
#Identification of GUID for the object
?WallInstance ifcowl:globalId_IfcRoot ?R .
?R express:hasString ?GlobalId .
Filter(?GlobalId="0g_QdKP4DAmhTDvOJ17p_H") .
#Identification of object name
OPTIONAL {
?WallInstance ifcowl:name_IfcRoot ?lable1 .
?lable1 express:hasString ?Name .}
#Identification of an ID tagged to the object.
OPTIONAL{
?WallInstance ifcowl:tag_IfcElement ?identifier1 .
?identifier1 express:hasString ?Tag .}
	?RelDefinesByProperty ifcowl:relatedObjects_IfcRelDefinesByProperties ?WallInstance .
	?RelDefinesByProperty ifcowl:relatingPropertyDefinition_IfcRelDefinesByProperties ?PropertySet
	#Identication of properties assigned to wall by PropertySets
	{
	?PropertySet ifcowl:hasProperties_IfcPropertySet ?Property .
	BIND(IRI(CONCAT(STR(?Property), "LOD200")) AS ?PropertyState).
	#Identification of Property name
	?Property ifcowl:name_IfcProperty ?PropertyIdentifier.
	?PropertyIdentifier express:hasString ?PropertyName.

3) Extraction of Object's construction details	
Optional{ ?RelAssociatesMaterial	

ifcowl:relatedObjects_IfcRelAssociates ?WallInstance .
#Extraction of object's Material Layerset
?RelAssociatesMaterial ifcowl:relatingMaterial_IfcRelAssociatesMaterial ?MaterialLayerSetUsage .
?MaterialLayerSetUsage ifcowl:forLayerSet_IfcMaterialLayerSetUsage ?MaterialLayerSet .
?MaterialLayerSet ifcowl:layerSetName_IfcMaterialLayerSet ?LayersetIdentifier.
?LayersetIdentifier express:hasString ?LayersetName .
# Extraction of object's Material Layer 	
?MaterialLayerSet ifcowl:materialLayers_IfcMaterialLayerSet/list:hasNext* ?MaterialLayerList .
?MaterialLayerList list:hasContents ?MaterialLayer.
BIND(IRI(CONCAT(STR(?MaterialLayer),"_Property")) AS ?layerProperty) .
BIND(IRI(CONCAT(STR(?MaterialLayer),"_PropertyState_LOD200")) AS ?layerPropertyState) .
# Identification of Material Thickness 	
?MaterialLayer ifcowl:layerThickness_IfcMaterialLayer ?LayerThicknessIdentity.
?LayerThicknessIdentity express:hasDouble ?LayerThickness.
#Identification of Material name
optional{
?MaterialLayer ifcowl:material_IfcMaterialLayer ?Material .
?Material ifcowl:name_IfcMaterial ?MaterialIdenitiy .
?MaterialIdenitiy express:hasString ?MaterialName .}
}
	#Identification of value assigned to property
	?Property ifcowl:nominalValue_IfcPropertySingleValue ?ValueIdentifier.
	?ValueIdentifier ?DataType ?PropertyValue .
	FILTER isLiteral(?PropertyValue) .
	}
	UNION
	#Identication of properties assigned to wall by QuantitySet
	{
	#Identification of Property name
	?PropertySet ifcowl:quantities_IfcElementQuantity ?Property.
	BIND(IRI(CONCAT(STR(?Property), "LOD200")) AS ?PropertyState) .
	?Property ifcowl:name_IfcPhysicalQuantity ?PropertyIdentifier .
	?PropertyIdentifier express:hasString ?PropertyName.
	#Identification of value assigned to property
	?Property ?Predicate ?ValueIdentifier .
	#Here ?Predicate indicates quantiy measure name. i.e. either area, volume, etc.
	Filter(?Predicate!=rdf:type && ?Predicate!=ifcowl:name_IfcPhysicalQuantity) .
	?ValueIdentifier ?DataType ?PropertyValue .
	FILTER isLiteral(?PropertyValue) .
	}

After the extraction of information from the BIM models (ifcOWL), construction queries are used to update the extracted information to DICL ontology. The table below illustrates these queries used for the data update process. Also, Table 21 illustrates a basic example of the constructed information according to the mentioned construction query within Table 20.

Table 20: The query for the construction of wall object data

CQ5: How is the BIM object is represented?	**CQ6:** How to represent multiple versions of information about the same object?
#Wall_Instance declaration	**#Property assignment**
?WallInstance a dice:BuildingObject . ?WallInstance dicl:hasID ?GlobalId; rdfs:label ?Name.	?Property a dicv:Property. ?WallInstance dicv:hasProperty ?Property. ?PropertyState a dicv:PropertyState. ?Property dicv:hasPropertyState ?PropertyState. ?Property rdfs:label ?PropertyName. ?PropertyState dicl:hasPropertyName ?PropertyName. ?PropertyState dicv:hasValue ?PropertyValue. ?PropertyState dicl:hasLODLevel diclvl:LOD_200.
#Wall construction details declaration	**#Layer Properties assignment**
?MaterialLayerSet a dicm:LayerSet . ?MaterialLayer a dicm:Layer . ?Material a dicm:Material . ?WallInstance dicm:hasLayerSet ?MaterialLayerSet. ?MaterialLayerSet rdfs:label ?LayersetName; dicm:hasLayer ?MaterialLayer. ?MaterialLayer dicm:hasMaterial ?Material. ?Material rdfs:label ?MaterialName.	?layerProperty a dicv:Property. ?layerPropertyState a dicv:PropertyState. ?MaterialLayer dicv:hasThickness ?layerProperty. ?layerProperty dicv:hasPropertyState ?layerPropertyState. ?layerPropertyState dicv:hasValue ?LayerThickness. ?layerPropertyState dicl:hasLODLevel diclvl:LOD_200.

Please note that here in this documentation prefixes used for the querying are not mentioned due to concern of document length. Also, the Class, object property, data property and instance declarations are as well not recorded in Table 20.

13.1.2 Filtering the data

The important segment in the demonstration process is the extraction or filtering of the information from the DICL ontology. This task is also performed as part of the ontology validation. As mentioned in the above subsection, the lifecycle ontology population is carried out by using SPARQL queries. The results from this ontology population process are mentioned in Table 21 below. The information in the below table only describes a sample of information about the single wall concerning the property representation and the construction details.

Table 21: Example 1 - Instances for the demonstration

```
@prefix dice: <https://w3id.org/digitalconstruction/0.5/Entities#> .
@prefix dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#> .
@prefix dicm: <https://w3id.org/digitalconstruction/0.5/Materials#> .
@prefix dicv: <https://w3id.org/digitalconstruction/0.5/Variables#> .
@prefix ifcowl: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD1/OWL#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix diclvl: <https://w3id.org/digitalconstruction/0.5/Levels#> .

#A-Box
<GlobalId/2af9a9d4-6443-4ac2-b74d-e584c11f4652#this>
                a                   dice:BuildingObject;
                rdfs:label          "Basic Wall:500+100:2145690";
                dicl:hasID          "0g_QdKP4DAmhTDvOJ17qPI";
                dicm:hasLayerSet    <IfcMaterialLayerSet_2009>;
                dicv:hasProperty    <IfcPropertySingleValue_2024>,
                                    <IfcPropertySingleValue_2033>,
                                    <IfcPropertySingleValue_2045>,
                                    <IfcPropertySingleValue_2046>,
                                    <IfcPropertySingleValue_8498>,
                                    <IfcPropertySingleValue_8624>,
                                    <IfcPropertySingleValue_8631> .

<IfcPropertySingleValue_2024>
        a                       dicv:Property;
        rdfs:label              "IsExternal";
        dicv:hasPropertyState   <IfcPropertySingleValue_2024_LOD300> .

<IfcPropertySingleValue_2024_LOD300>
        a                       dicv:PropertyState;
```

```
                     dicl:hasLODLevel          diclvl:LOD_300;
                     dicl:hasPropertyName      "IsExternal";
                     dicv:hasValue             true .

<IfcMaterialLayerSet_2009>
                     a                         dicm:LayerSet;
                     rdfs:label            "Basic Wall:500+100";
                     dicm:hasLayer             <IfcMaterialLayer_2004>,
                                               <IfcMaterialLayer_2007>,
                                               <IfcMaterialLayer_2008>,
                                               <IfcMaterialLayer_8390> .

<IfcMaterialLayer_2004>
                     a                         dicm:Layer;
                     dicm:hasMaterial          <IfcMaterial_1939>;
                     dicv:hasThickness         <IfcMaterialLayer_2004_Property> .

<IfcMaterial_1939>   a                         dicm:Material;
                     rdfs:label            "Cement plaster" .

<IfcMaterialLayer_2004_Property>
                     a                         dicv:Property;
                    dicv:hasPropertyState  <IfcMaterialLayer_2004_PropertyState_LOD300> .

<IfcMaterialLayer_2004_PropertyState_LOD300>
                     a                         dicv:PropertyState;
                     dicl:hasLODLevel          diclvl:LOD_300;
                     dicv:hasValue             2.0E1 .

diclvl:LOD_300       a                         dicl:LODLevel;
                      rdfs:label               "LOD 300" .
```

In this subsection, this stored information is queried or filtered using also a set of SPARQL queries. These query profiles along with the results are elaborated in Table 22 and Table 23. Both the queries mentioned in the above tables are querying the information on a generic basis without concerning the specific LOD level. But this data extraction process can be limited to the specific LOD level by using the “FILTER” function from the SPARQL query. This FILTER function is already mentioned in the above query profiles but is disabled for querying the complete information.

Table 22: Object property extraction

CQ7: How the object properties and values for a specific LOD level are defined?

CQ7.1:- Extraction of object properties along with their LOD level specification

```
Select Distinct ?ObjectName ?PropertyName ?PropertyValue ?LODLevel
where {
    ?BuildingObject dicv:hasProperty/dicv:hasPropertyState ?PropertyState .
    ?BuildingObject rdfs:label ?ObjectName .
    ?PropertyState dicl:hasPropertyName ?PropertyName .
    Filter(?ObjectName="Basic Wall:500+100:2145690").
   {?PropertyState dicv:hasValue ?PropertyValue;
                   dicl:hasLODLevel ?Level.}
   ?Level rdfs:label ?LODLevel.     }
```

Query Results

	ObjectName	**PropertyName**	**PropertyValue**	**LODLevel**
1	"Basic Wall:500+100:2145690"	"IsExternal"	"true"^^xsd:boolean	"LOD 200"
2	"Basic Wall:500+100:2145690"	"LoadBearing"	"false"^^xsd:boolean	"LOD 200"
3	"Basic Wall:500+100:2145690"	"ExtendToStructure"	"false"^^xsd:boolean	"LOD 200"
4	"Basic Wall:500+100:2145690"	"Reference"	"500+100"	"LOD 200"
5	"Basic Wall:500+100:2145690"	"Description"	"Exterior Wall"	"LOD 200"
6	"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"0.252765957446808"^^xsd:double	"LOD 200"
7	"Basic Wall:500+100:2145690"	"Width"	"0.6"^^xsd:double	"LOD 200"
8	"Basic Wall:500+100:2145690"	"Height"	"3.8"^^xsd:double	"LOD 200"
9	"Basic Wall:500+100:2145690"	"NetVolume"	"41.0625830959226"^^xsd:double	"LOD 200"
10	"Basic Wall:500+100:2145690"	"GrossSideArea"	"2.28"^^xsd:double	"LOD 200"
11	"Basic Wall:500+100:2145690"	"GrossVolume"	"1.368"^^xsd:double	"LOD 200"
12	"Basic Wall:500+100:2145690"	"GrossFootprintArea"	"13.7084429199795"^^xsd:double	"LOD 200"
13	"Basic	"NetSideArea"	"68.4376384932042"^^xsd:dou	"LOD 200"

	Wall:500+100:2145690"		ble	
14	"Basic Wall:500+100:2145690"	"Length"	"22.8474048666326"^^xsd:double	"LOD 200"
15	"Basic Wall:500+100:2145690"	"IsExternal"	"true"^^xsd:boolean	"LOD 300"
16	"Basic Wall:500+100:2145690"	"ExtendToStructure"	"false"^^xsd:boolean	"LOD 300"
17	"Basic Wall:500+100:2145690"	"LoadBearing"	"false"^^xsd:boolean	"LOD 300"
18	"Basic Wall:500+100:2145690"	"Reference"	"500+100"	"LOD 300"
19	"Basic Wall:500+100:2145690"	"ThermalTransmittance"	"0.262485417476807"^^xsd:double	"LOD 300"
20	"Basic Wall:500+100:2145690"	"FireRating"	"REI60"	"LOD 300"
21	"Basic Wall:500+100:2145690"	"Description"	"Exterior Wall - Brick"	"LOD 300"
22	"Basic Wall:500+100:2145690"	"Width"	"640."^^xsd:double	"LOD 300"
23	"Basic Wall:500+100:2145690"	"Height"	"3900."^^xsd:double	"LOD 300"
24	"Basic Wall:500+100:2145690"	"GrossSideArea"	"2.496"^^xsd:double	"LOD 300"
25	"Basic Wall:500+100:2145690"	"NetVolume"	"45.0582231789041"^^xsd:double	"LOD 300"
26	"Basic Wall:500+100:2145690"	"NetSideArea"	"70.4034737170378"^^xsd:double	"LOD 300"
27	"Basic Wall:500+100:2145690"	"GrossVolume"	"1.59744"^^xsd:double	"LOD 300"
28	"Basic Wall:500+100:2145690"	"GrossFootprintArea"	"14.8026213279241"^^xsd:double	"LOD 300"
29	"Basic Wall:500+100:2145690"	"Length"	"23129.0958248814"^^xsd:double	"LOD 300"

Table 23: Construction details Extraction

CQ7: How the object properties and values for a specific LOD level are defined? **CQ7.2:- Extraction of object construction details specified in different LOD levels**

Select Distinct ?ObjectName ?Layerset ?MaterialLayer ?Thickness ?LODLevel
where {
?BuildingObject dicm:hasLayerSet ?Layerset .
?Layerset dicm:hasLayer ?ObjectLayer .
?BuildingObject rdfs:label ?ObjectName .
Filter(?ObjectName="Basic Wall:500+100:2145690").
?ObjectLayer dicm:hasMaterial ?Material.
?Material rdfs:label ?MaterialLayer.
?ObjectLayer dicv:hasThickness/dicv:hasPropertyState ?Property.
{?Property dicv:hasValue ?Thickness;
dicl:hasLODLevel ?Level.}
?Level rdfs:label ?LODLevel
}

Query Results

	ObjectName	**Layerset**	**MaterialLayer**	**Thickness**	**LODLevel**
1	"Basic Wall:500+100:2145690"	IfcMaterialLayerSet_2009	"Cement plaster"	"20."^^xsd:double	"LOD 300"
2	"Basic Wall:500+100:2145690"	IfcMaterialLayerSet_2009	"Styrofoam"	"100."^^xsd:double	"LOD 300"
3	"Basic Wall:500+100:2145690"	IfcMaterialLayerSet_2009	"Brick"	"500."^^xsd:double	"LOD 300"
4	"Basic Wall:500+100:2145690"	IfcMaterialLayerSet_2009	"Plaster"	"20."^^xsd:double	"LOD 300"
5	"Basic Wall:500+100:2145690"	IfcMaterialLayerSet_2012	"Brick Wall"	"0.5"^^xsd:double	"LOD 200"
6	"Basic Wall:500+100:2145690"	IfcMaterialLayerSet_2012	"Thermal isolation"	"0.1"^^xsd:double	"LOD 200"

In the development of Digital Construction Lifecycle ontology, the elements related to the building lifecycle process (activity, BLS, agent, etc.) is also considered. However, this demonstration is not focused on those process elements. Nevertheless, the demonstration of these process elements can be performed by using the instances represented in Figure 24, Figure 25 and Figure 26. For more understandings, Table 24 is developed with respect to a simple process concerning a set of activities and the involved agents in it.

Table 24: Example 2 - Instances for the demonstration

```
@prefix dice: <https://w3id.org/digitalconstruction/0.5/Entities#> .
@prefix dicl: <https://w3id.org/digitalconstruction/0.5/Lifecycle#> .
@prefix dicm: <https://w3id.org/digitalconstruction/0.5/Materials#> .
@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 dcterms: < http://purl.org/dc/terms/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

#For T-BOX taxonomy and the relations are clearly illustrated in figure 7.
#A-Box
<3.5_DetailedDesign>          rdf:type              dicl:BLStage ;
                                   dicl:hasActivity      <125_BuildingPermissions> .

<125_BuildingPermissions>     rdf:type              dicp:Activity ;
                                   dica:hasAgent         <Project_Leader> ;
                                   dicp:hasObject        <GlobalId/2af9a9d4-6443-4ac2-b74d-
e584c11f4652> ;
                                   dcterms:description   "documents required for building permit
applications,
draft and submit applications".
<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> ;
                                        dcterms:description   "architectural plans and documentation
describing the project to a level of detail as required for Planning or Building permit applications".
<124_TechnicalDetails>        rdf:type                dicp:Activity ;
                                        dica:hasAgent           < Building_Services_Designer> ;
                                        dicp:hasObject          <GlobalId/2af9a9d4-6443-4ac2-b74d-
e584c11f4652> ;
                                        dcterms:description  "additional technical documentation from
technical specialist consultants; such as acoustic, thermal, fire safety, environmental and other
appraisals as required by applicable legislation ".
```

As a conclusory statement, the above subsections of this demonstration appendix provides comprehensive discussions on representing available BIM data in different LOD levels. More specifically these discussions are extended on the representation of Building object properties and construction details. Similarly, the elaborated explanations on the adopted LOD frameworks and their ontological representations along with the applicability are provided in earlier sections of this document. The demonstration process clearly illustrates the applicability of the developed lifecycle ontology concerning the representation of BIM data and filtering the data based on the required Level of Detail.