BIM4EEB

BIM based fast toolkit for Efficient rEnovation in Buildings

# D3.4 An Ontology to Represent Renovation Workflows Including BIM Change Management

## – Contexts, Variables, Entities, Processes, Agents, and Information

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 |

# D3.4 An ontology to represent renovation workflows including BIM change management – Contexts, Variables, Entities, Processes, Agents and Information

| | |
|---|---|
| **Work Package** | WP3 |
| **Lead Partner** | VisuaLynk |
| **Contributing Partner(s)** | VTT |
| **Dissemination Level** | Public |
| **Type** | Other |
| **Due date** | 31/12/2019 |
| **Updated version 3.0 submission date** | 07/02/2022 |
| **Version** | 3.0 |

# DOCUMENT HISTORY

| Version | Date | Comments | Main Authors |
|---|---|---|---|
| 0.1 | 25.10.2019 | Table of content created | Seppo Törmä (VisuaLynk) |
| 0.2 | 11.11.2019 | First draft version completed | Seppo Törmä (VisuaLynk) |
| 0.3 | 05.12.2019 | Content about domain theories | Seppo Törmä (VisuaLynk) |
| 0.4 | 08.12.2019 | Version for internal review | Seppo Törmä (VisuaLynk) |
| 1.0 FINAL | 19.12.2019 | Final version for submission | Seppo Törmä (VisuaLynk) |
| 1.1 | 16.04.2021 | Second version started | Seppo Törmä (VisuaLynk) |
| 1.2 | 11.05.2021 | Segmentation diagram, class overview diagram, objectified properties (model and alignment), constraints diagram | Seppo Törmä (VisuaLynk) |
| 1.3 | 12.05.2021 | Activities, agents, information. Removed obsolete contents. | Seppo Törmä (VisuaLynk) |
| 1.4 | 13.05.2021 | Finalized for internal review | Seppo Törmä (VisuaLynk) |
| 1.5 | 25.05.2021 | Example of data organization added | Seppo Törmä (VisuaLynk) |
| 1.6 | 27.05.2021 | Revisions throughout the text. Better diagrams for background section. | Seppo Törmä (VisuaLynk)<br>Kostas Tsatsakis (S5) |
| 1.9 | 27.05.2021 | Ready for finalization | Seppo Törmä (VisuaLynk) |
| 1.10 | 31.05.2021 | Review of PoliMi's team including PC. Plagiarism check. | Polimi |
| 2.0 | 30.06.2021 | Answers to monitors comments, and the last revisions to the contents. Approval of WP3 leader. Approval of PC. To be submitted. | Seppo Törmä (VisuaLynk) |
| 2.1 | 09.11.2021 | New version created. | Seppo Törmä (VisuaLynk) |
| 2.2 | 16.11.2021 | Added a subtitle to the deliverable, listing the ontology modules | Seppo Törmä (VisuaLynk) |
| 2.3 | 01.12.2021 | Completed the requested changes | Seppo Törmä (VisuaLynk) |
| 3.0 | 15.01.2022 | Changes due to the internal review.<br>As a result the WP Leader approved the developed version. | Janakiram Karlapudi (TUD)<br>Karsten Menzel (TUD)<br>Meherun Lucky (PoliMI)<br>Seppo Törmä (VisuaLynk) |
| 3.0 FINAL | 31.01.2022 | Final version, after final and formal review and plagiarism check. | Bruno Daniotti, Martina Signorini (PoliMi), Gabriele 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

The ongoing digitalization of construction industry is changing the methods for managing renovation projects. Designs will be increasingly produced as BIM models, procurement and construction management will use information systems, the workers at construction site have mobile devices, and increasing amounts of sensors and imaging data is coming available. These systems provide more concrete and timely observations about the status of models, shipments, building objects, and so on. This deliverable provides an ontology for renovation processes that makes it possible to represent the connection between activities and their ingredients that can be directly observed, thus paving a way to systems that can facilitate real-time management of renovation projects. The ontology builds upon the standards that address the digital exchange of information in construction domain (ISO 16739 IFC, ISO 19650 BIM/IM, ISO 21597 ICDD), and ontology standards (ISO 21838 BFO) as well as other established ontologies.

This deliverable describes a set of interlinked ontology modules belonging to Digital Construction Ontologies (DiCon): Contexts, Variables, Entities, Processes, Agents and Information. Together these modules enable the representation of renovation processes to support construction production management in renovation projects. The ontologies will support ordinary construction management practices and can therefore provide the conceptual foundation even for projects that do not employ digital technologies or advanced construction management methods. At the same time, they support the utilization of digital information from sensors, mobile devices, and other relevant information systems. The ontologies take into account the various dependencies and constraints in construction phase by covering the entities faced in construction management - building objects, information objects, delivery teams, task teams, locations and workspaces, equipment, materials, and external conditions – and their connections to activities. The ontology also addresses the dynamic evolution of information during project execution: designs go through several levels of development, planning is carried out at multiple levels, and plans are also periodically adjusted. Information can be represented at multiple different contexts according to ISO 19650 and additional metadata can be provided about information, supporting BIM change management.

The ontology uses the terminology of ISO 19650 and is aligned with IFC (ISO 16739) and BFO (ISO/IEC 21838) as well as with many reference ontologies, such as SSN/SOSA and Saref Core/Buildings for sensor data, OWL-Time for temporal entities, QUDT for units of measurement, FOAF and Org for agents and organizations, and PROV-O for provenance information.

# PUBLISHING SUMMARY

The ongoing digitalization of construction industry is changing the methods for managing renovation projects. This deliverable describes a set of ontology modules – for Contexts, Variables, Entities, Processes, Agents, and Information, belonging to Digital Construction Ontologies (DiCon) – for the representation of processes in the delivery phase of renovation projects. They make it possible to represent the connections between activities and the various kinds of entities that can be directly observed – building objects, information objects, delivery teams, task teams, locations and workspaces, equipment, materials, and external conditions – thus paving a way to systems that can related real-time sensors data to renovation activities. The ontology also addresses the dynamic evolution of information during project execution: information can be represented at multiple different contexts and additional metadata can be provided about the information, supporting change management. The ontology builds upon the standards that address the digital exchange of information in construction domain (ISO 16739 IFC, ISO 19650 BIM/IM, ISO 21597 ICDD), ontology standards (ISO 21838 BFO), and other established ontologies.

# TABLE OF CONTENTS

# LIST OF FIGURES



# LIST OF TABLES

# 1 Introduction

Workflow management concerns the supervision and monitoring of work processes. The focus is not on the specifics of individual activities – to whose execution there are numerous existing point solutions – but on the logistical aspects of processes: the flow of work through organization (Van der Aalst, 1998). The central aim is the automatization of processes combining human and machine-based activities, especially those associated with information systems (Hollingsworth, 1995). Workflow management can be regarded as *automatization of the logistic aspects of processes*: guiding the execution of processes step-by-step while taking care that the preconditions of each activity are satisfied, that the necessary ingredients are available to enable its execution, and that its results are properly forwarded to subsequent activities.

Workflow management has originally been used mostly for the execution of business processes in administrative contexts, often under the title of *business process management*. Some examples of such flows of work are claims processing, customer support, media publishing, request/approval processes and workforce recruitment. In such domains, the important activity ingredients are agents that should execute activities and information entities consumed or produced by activities. Advanced workflow management systems inform personnel of the tasks they can or should execute and route the relevant information to each activity execution.

## 1.1 Workflow management in the delivery phase of renovation

The perspective adopted in this work is the following: *due to the recent progress of digitalization, the concept of workflow management can be applied also to construction processes of renovation projects*. Since dynamic and almost instantaneous data from sensors and mobile devices about physical entities is increasingly available, those physical entities can be monitored in the same way as information entities in traditional workflow management, and the relevant personnel can be informed about the activities than can or should be executed and provided with the appropriate information through digital channels. Currently, such sensor data can already be collected from positions of people, equipment, building objects and materials, the occupancy of locations, and the temperature and humidity conditions of work locations.

Already for some time, advanced renovation and construction projects have been based on digital design and planning systems. There is scanning and mapping data about existing buildings, BIM-based design models, product data from various repositories, data about procurement and supply-chain processes, construction plans and schedules. The dynamic data about physical entities – status updates from mobile devices, and sensor-based monitoring data – can be interpreted in the context of these models, to enhance predictions and decisions.

The BIM4EEB project has mapped the managerial and design-related processes of renovation projects including the detailed information flows (BIM4EEB-D21, 2019). In the area of construction management, the complex variety of dependencies in the physical stage of construction projects – ranging from building objects, material batches and labor crews to workspaces, equipment, information, and environmental conditions – have been identified in the past research (Koskela 1999, 2000) (Garcia-Lopetz 2017). This deliverable provides the representational structures for capturing the dependencies caused by these different ingredients of activities.

Despite the new opportunities created by the emerging information sources, many challenges remain in the proper management of dependencies in renovation processes. Firstly, the large number and heterogeneity of dependencies make plans fragile: a failure to satisfy any one of the dependencies usually makes an activity inexecutable, therefore requiring additional setup work and causing delays rippling down to subsequent activities. Secondly, there are several parallel threads of activity whose use of shared resources needs to be carefully coordinated. Delays in one of the threads can cause conflicts with the activities in the other threads. Thirdly, the execution of an activity can fail, perhaps only partially, causing both operational issues and quality deviations. Consequently, there is a need to monitor and manage complex dependencies of activities and to maintain constant readiness to revise plans and schedules.

The purpose of this report is to describe a set of ontology modules of Digital Construction Ontologies that allow the representation of dependencies in renovation processes and enable the monitoring of the execution based on a range of different types of dynamic progress information.

To capture these representational needs, several different threads of previous work need to be integrated. Firstly, the definition of processes is a result of various planning and scheduling functions, where the activities, their constraints, timing and resource assignments are specified. There are many models of activities ranging from Critical Path Method (Kelley 1959) that capture precedence relations between activities to Planning Domain Definition Languages (Fox 2002, 2003) and state-variable representations (Ghallab, 2016) that capture the relations of activities to the states of entities in the underlying domain. Secondly, the aspects of underlying domain that need to be covered when defining processes have been identified in construction management research (Koskela 1999, 2000) (Garcia-Lopetz, 2017). There are modern construction practices – such as Last Planner method (Ballard, 2000), Location-Based Management System (Kenley, 2006) (Seppänen, 2010, 2014b), or Takt Time planning (Heinonen, 2016) (Seppänen, 2014a) – that aim to capture or simplify the dependencies to ensure a smooth and efficient execution of processes. Thirdly, there are various detailed models related to particular dependencies, such as workspace planning (Akinci, 2002) and material kitting, that should be possible to integrate in the whole. Finally, as sensors are emerging in the construction and renovation sites, it is important to be able to relate their observations to construction processes. There are many existing ontologies for representing monitoring systems and sensor observations, such as the Semantic Sensor Network Ontology (Haller, 2017) or Saref Core/Building (Poveda-Villalón, 2018).

A particular challenge in any process related ontology is the dynamic nature of the domain: the situation changes constantly (BIM4EEB-D31, 2019). There is both accumulation of actual information about the execution, similar than what could be included log files or journals, and accumulation of modal entities, future-oriented information that specifies the goals of the execution – such as renovation scenarios, BIM models, drawings, master plans, breakdown structures – often having multiple levels or versions. Renovation and construction projects have a complex structure of such contexts: there is one set of contexts for design and another for plans, and there are dependencies between these sets. In the ontologies the temporal evolution of information is modelled according to the guidelines of established ontologies such as SSN/SOSA, Saref Core, QUDT; the modal contexts are modelled according to the ISO 19650 concepts of information model and information container.

Most of the above-mentioned preceding work has been done in the context of either project management or construction management, not specifically on renovation management. However, it is directly applicable to renovation projects, since renovation projects can be regarded as construction projects in the context of an existing building (BIM4EEB-D31, 2019). The relative distribution of work amounts in different disciplines can be different from renovation to new construction – somewhat smaller average project size, more controlled demolition work, less excavation and frame construction, relatively more emphasis on building systems, outer shell, and indoor construction – but the underlying logic is the same. Also, there are often additional information flows related to an existing building, its performance profile, and existing owners and occupants. However, at the concrete level the elementary tasks and processes are mostly quite similar; therefore, the results and models are applicable also to renovation contexts.

In summary, the objective of this report is to *define a set of ontology modules to represent the flow of work among delivery phase activities, related entities, and information models in renovation projects*. The formalization will be done in Web Ontology Language (OWL). The definitions are published as part of Digital Construction Ontologies, consisting of the following interrelated ontology modules: Variables, Contexts, Entities, Processes, Agents, and Information. The modules cover different aspects of renovation activities, the entities that define the flow of work between activities, and the information management concerning the plans governing the execution of activities. The ontology should enable the representation of the ingredients of activities, dependencies between activities (precedence and decomposition), give operational guidance to actors (what to do next), include information of what is needed for execution of each activity (resources such as labour, materials, or equipment), provide the status of execution, and information about its deviation from the plan.

## 1.2 References to other activities and to the state of the art

Following the principle of reuse of existing work, the development on the ontologies in BIM4EEB is built upon the ontology work a national Finnish research project Diction that was a collaborative research and development project on real-time situational awareness in construction. It was carried out during 2018-2020 and funded by Business Finland (previously Tekes). In the early phases of Diction, an ontology for construction processes was developed. The project had a strong participation from construction and renovation industry, and ontology development was based on a series of industry-driven workshops during 2018 to map the shared concepts required for construction management and situational awareness. Both researchers and practitioners of advanced construction management methods, as well as related tool developers, were participating in the process. A set of ontology modules – Activities, Organizations, Monitoring, and Planning – were specified in Diction during 2018 and early 2019, based mostly on input from industrial and academic partners with expertise on construction management.

As reported in the deliverable D3.1 (BIM4EEB-D31, 2019), the state-of-the-art study of BIM4EEB identified the Diction ontologies as the most relevant and ambitious construction workflow ontology. It was consequently agreed in early 2019 that since Diction was already completing the ontology work and moving to application phase, BIM4EEB could continue the development of the initial ontologies of Diction and the resulting enhanced ontologies would also be shared with the Diction project that could use the enhanced ontologies in the case studies and applications developed in the project. The objective was to reuse the existing work, to achieve higher quality of the end results, and to get larger unified community behind the ontologies. As a result of the state-of-the-art study and analysis of use cases conducted in the BIM4EEB Task 3.1, the development focus turned to standards compliance, extensions of the ontologies, alignment with existing ontologies, formal aspects of ontology definition, documentation, and validation in the applications developed in BIM4EEB.

In BIM4EEB the previously defined ontology modules were refactored and eventually completely replaced by the following set of Digital Construction Ontologies modules: Variables, Contexts, Entities, Processes, Agents and Information. Particular emphasis was placed on the compliance with relevant standards, such as ISO 16793, ISO 19650, ISO 21597, and ISO/IEC 21838. Moreover, a set of new modules were specified, as a result of work in Task 3.2, Task 3.3 and Task 3.5: Building Materials, Occupant Behavior, Indoor Air Quality, Energy Systems, and Building Acoustics. Later on, these modules were reorganized as the following modules: Materials, Lifecycle, Occupancy, and Energy. New vocabularies were also created: Units, Levels, and Stages. Ontologies were reorganized based on the vertical and horizontal segmentation approach specified in the documentation of the SSN ontology, and the external references were made explicit in separate alignment modules. Ontology versioning was implemented to support long-term maintenance. The documentation was generated using WiDoCo/pyLODE and self-implemented tools, and ontology diagrams were redrawn according to the Chowlk notation. Furthermore, the alignments were defined with respect to relevant established ontologies: QUDT, OPM, OWL-Time, wgs84_pos, PROV-O, SSN/SOSA, FOAF, Org, Saref, ifcOWL, BOT, RealEstateCore, ICDD, and DCAT2, including the RDF reification terminology.

In the end, in BIM4EEB all the specific ontology definitions and diagrams made in Diction were replaced with new ones and completely reorganized but the original conceptual work based on industry contribution related to construction workflows, as well as related use case specifications, have provided invaluable understanding and guidance to the ontology definition process in BIM4EEB. Equal results would not have been possible if the ontology work in BIM4EEB had started from scratch. Overall, this experience provides further validation to the generally accepted view that wide collaboration and reuse of existing work in ontology development are highly beneficial.

The ontology modules defined in this deliverable – Variables, Contexts, Entities, Processes, Agents, and Information – form the basis of the Digital Construction Ontologies (DiCon). Other modules are built on top of them. The ontology modules provide a standards compliant model of the central terminology defined in ISO 19650 BIM/IM. The modules are aligned with ISO 16793 IFC, ISO 21597 ICDD Container ontology and ISO/IEC 21838 BFO. The ontologies have been published in an open manner to invite further contributions from subsequent projects and efforts. The alignment of DiCon ontologies with the ontology work carried out in the Diction project is provided in the Deliverable 3.6, Appendix I (BIM4EEB-D36, 2022).

The work in the Task 3.4 provides input to the Task 3.6 that presents the overall framework of Digital Construction Ontologies developed in the Work Package 3 of BIM4EEB project. The harmonized model of Task 3.6 is utilized by the tools developed in WP6 and WP7 of BIM4EEB such as BIMEaser, BIMPlanner and BIM4Occupants.

## 1.3 Innovative results and progresses

The deliverable provides the ontology definitions of the entity types that are relevant to construction management in renovation projects. It is based on the state-of-the-art models of construction management and integrates the approaches of different standards. The following results and progresses can be identified regarding the ontologies produced:

- A Linked Data model for ISO 19650 has been defined in the Contexts and Information ontologies. It covers the ISO 19650 concepts of Information Model, Information Container, Federation, and Information Container Breakdown Structure. Each Information Container is modelled in a Linked Data manner: its content is represented as RDF in its own named graph in an RDF dataset. These definitions for Information Models and Information Containers (and their subtypes of Datasets and Linksets) have been aligned with ISO 21597 Container ontology, again assuming that the relevant contents of the Documents defined in the ICDD Container can be converted into an RDF graph. Furthermore, the alignment of these concepts with DCAT2 Catalog ontology is provided. The benefit of these concepts is to give an organized and standards compliant structure for the complex information entities produced in the renovation projects, including BIM models with LOD levels to breakdowns structures and operational plans with different levels and versions.
- The Variables ontology defines a model for properties that can have variable values over the time. The Variable ontology support both a constraint-based view to the properties as well as observation-based view. The core concepts are aligned with QUDT, SSN/SOSA and Saref Core. The alignment with the RDF reification terminology makes the Variables ontology compliant with the upcoming RDF*/SPARQL* specifications that is already implemented by many RDF databases.
- Entities ontology provides a superstructure of the ontology aligned with ISO/IEC 21838 BFO, making Digital Construction Ontologies the first ontology in construction domain to comply with BFO. The significance is that BFO compliance make it easier to align DiCon with many other large BFO-based ontology efforts, such as the Industrial Ontologies (in the manufacturing domain), European Materials Modeling Ontology EMMO (in the materials domain) and Open Energy Ontology OEO (in the energy domain).
- The Agents ontology provides definitions for ISO 19650 terms of Team, Delivery Team, Task Team, Appointment, Appointing Party, Appointed Party, Lead Appointed Party, as well as legal concepts, allowing the proper definition of Built Asset Owner.
- The set of Digital Construction Ontology modules defines the activity-flow relations defined in the construction management research (Koskela 200) that make it possible to manage the complex

set of dependencies between activities.

## 1.4 Structure of the deliverable

The structure of the deliverable is the following. Section 2 contains the background of representing delivery phase activities, entities and information models, referring to related work, existing standards and established ontologies. Section 3 provides the requirements definition, using competency questions for functional requirements, and defining the important non-functional requirements and non-requirements. Section 4 outlines the ontology modules, starting from an overview and progressing to detailed discussion of main objects, reuse of ontologies and alignments. Section 5 presents the example of how ontology is to be used. The alignments and validation of the Digital Construction Ontologies is presented in Deliverable 3.6.

All ontology diagrams in this deliverable are drawn according to the Chowlk visual notation[1] using the draw.io[2] diagramming software with Chowlk library definitions.

[1] Chowlk visual notation: https://chowlk.linkeddata.es/chowlk_spec
[2] Draw.io: https://app.diagrams.net/

# 2 Ontology background

This section describes the background for representing the flow of work in the delivery phase of a renovation project.

The central concept in the domain of construction workflows is that of an *Activity*, defined in this work as follows:

> *Activity* is an *intentional process, with a start and end, carried out by an agent*.

The concept of an activity covers different kinds of purposeful and value-adding processes in renovation projects or built asset management. For instance, all the following are examples of activities, large and small:

> Renovation project; Designing a building; Procuring a product; Approving work; Transporting materials to a construction site; Painting; Pouring concrete; Building and dismantling formwork; Inspecting; Excavating: Erecting scaffoldings; Changing windows;
> Making energy simulation; Preparing a plan; Creating an issue.

*Activity* is used as a generalization of a range of other familiar concepts, such as a *task*, *operation*, *phase*, *project*, and so on. However, all processes occurring over time are not activities: outside of the concept of an activity are *events* that happen without a known intention, *behavior processes* that do not have identifiable start and end times, or *service provision* that is an ongoing process of reactive responses to requests.

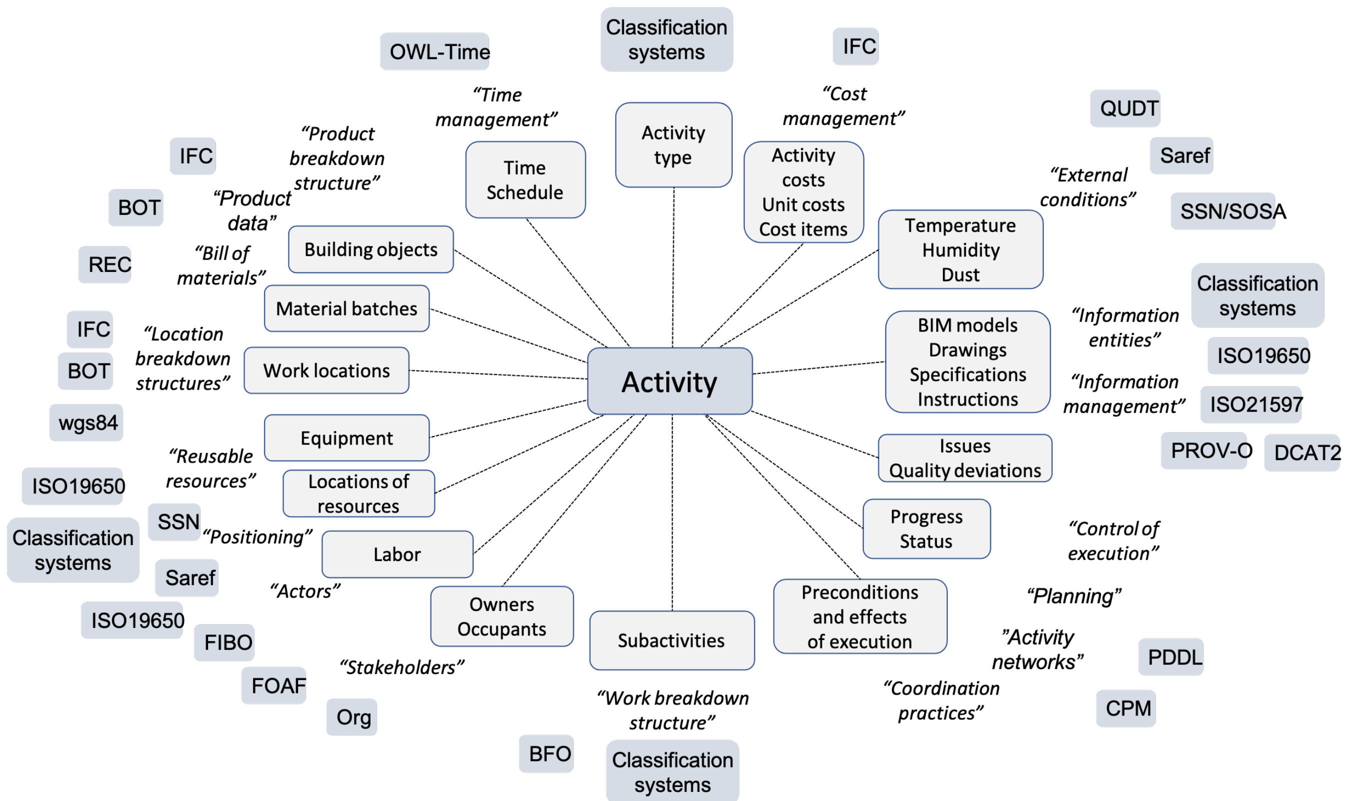


**Figure 1: Different aspects of activities with relevant ontologies and models**

Activities have a range of different relations to other entities in construction and renovation domain, as shown in Figure 1. The ontologies and models related to these aspects, as identified in the Deliverable 3.1 (BIM4EEB-D31, 2021), are shown at the outer edge of the figure. Many of the aspects describe connections of an activity to its ingredients; the use of the same ingredient instance is a source for dependencies between activities. Therefore, the representation of the linking of activities to their ingredients is an essential enabler for the informed management of activity dependencies, and consequently, the flow of work.

Figure 1 shows, in a summary, that activities can be of different types, relating them to various classification systems in the construction domain (such as Uniclass, Omniclass, CoClass, Talo2000, bSDD). Activities happen in time and have schedules and actual execution times (related to OWL-Time), and they have cost associated with them (related to IFC cost model). Activities can be recursively decomposed into subactivities from the levels of a project all the way down to individual tasks (related to, for example, BFO's and IFC's decomposition relations).

The minimum requirement for the representation of workflows in BIM-based projects is that the activities are connected to the building objects defined in BIM models (available in IFC or BOT). In addition to building objects, however, activities are also associated with material batches, work locations, and equipment (related to IFC, BOT, and RealEstateCore). They require labour resources for execution (related to ISO 19650, FOAF, and Org). The external conditions – temperature, humidity, dustiness – may need to be in proper range to allow the execution (related to SSN/SOSA, Saref, QUDT). The positions of entities can be tracked with positioning technology (related to wgs84, SSN/SOSA, Saref). In the planning phase, the preconditions and effects of activities need to be considered for proper sequencing of the activities to form executable activity networks (related to PDDL, I-N-OVA, CPM or PERT).

The plan and designs – including BIM models – need to have a proper organization in a renovation project (related to ISO 19650 and DCAT2) and exchanged of interlinked designs and plans needs to be supported (related to ISO 21597). At the execution time the preconditions need to be checked and managed in a proper way, a process that is systemized, for instance, in the Last Planner method. During the execution time various issues may be encountered, including quality deviations. The execution of activities is related to stakeholders, parties that are not directly participating in the execution but who have an interest on it (related to FOAF, Org, FIBO). The different aspects and ontologies need to be organized and related according to sensible fundamental categories – such as material entities, spatial entities, temporal entities, information entities, roles, capabilities, and so on – in a principled manner, where a standards compliant approach is needed (related to BFO).

The concepts to describe processes in the delivery phase of a renovation project and create dependencies that determine the flow of work, occur in a variety standards and ontologies. This deliverable aims to bring together these different models for the representation of renovation projects.

## 2.1 Use cases

The use cases shown in Table 1 have been identified in the Deliverable 3.1 (BIM4EEB-D31, 2021) regarding construction workflows and construction management in renovation projects. They concern the development of the master plan for a renovation project, the definition of tasks to be executed taking into consideration of the renovation measures and structure of the existing building, the control and monitoring of the execution and its progress, plan revisioning and related information management challenges, week planning to select the executable activities to execution, and coordination and communication with occupants and owners about the activities to execute.

**Table 1: Use cases for construction workflow management**

| **Use Case** | **1 Master planning** |
|---|---|
| **Data required** | • Continuation plan from the initiative phase<br>• Renovation measures<br>• Planning principles and execution strategies |

| | |
|---|---|
| | • Expected budget requirements<br>• Data from past projects |
| **Actions** | • The development of the master plan of a project taking into account the general planning principles, execution strategies (e.g., types of contracts, subcontracting, prefabrication), renovation measures, and budget constraints.<br>• Initial estimation is based on the scope of the project, in revised versions on quantity surveys. |
| **Data produced** | • Master plan, including:<br>  • Project team structure<br>  • Work breakdown structure<br>  • Target schedule for activities<br>  • Allocation of responsibilities |
| **Use Case** | **2 Task planning** |
| **Data required** | • Design information: BIM models<br>• Master plan / week plan<br>• Project team composition<br>• Estimates of activity size |
| **Data produced** | • Development of a location breakdown structure<br>• Creation of location-specific activity decomposition<br>• Requirements/assignments for exchangeable resources (types/capabilities of equipment and agents)<br>• Connections of building objects to activities |
| **Outcome** | • Concrete, executable activities including:<br>  ▪ Connections of activities to building objects, locations, equipment, and agents<br>  ▪ Activity durations and costs |
| **Use Case** | **3 Progress coordination** |
| **Data required** | • Master plan / week plan<br>• Status updates / sensor data<br>• Procurement data (types/instances)<br>• Data about product-process connections |
| **Actions** | • Providing operational guidance to actors based on the plans (what to do next/ what to do today)<br>• Monitoring the execution of activities: start/end, positions of entities, issues encountered<br>• Detection of deviations from plans<br>• Gathering and recoding enhanced data during execution |
| **Data produced** | • Enhanced product and process information<br>• Detected issues, deviations, or other needs for plan revision |
| **Use Case** | **4 Plan revisioning** |
| **Data required** | • Master plan / week plan<br>• Evolving data about entities (new versions/levels of BIM models, procurement data, appointments)<br>• Previously planned revision schedule (e.g., time to create the next week's plan)<br>• Detected issues, deviations, or other needs for plan revision |
| **Actions** | • Creation of a new version of a plan<br>• Approving the plan for execution<br>• Managing the information so that the content of the new plan becomes basis of execution |
| **Data produced** | • The new plan content<br>• New information container for plan<br>• Updated information model to activate the plan |
| **Use Case** | **5 Week planning** |
| **Data required** | • Detailed designs for construction<br>• Location-breakdown structure |

| | |
|---|---|
| | • Master plan and previous week plans<br>• Monitoring data about current week's activities<br>• Connections of activities to building objects, locations, equipment, and agent<br>• Activity durations |
| **Actions** | • The activities to be executed during the next week are selected and scheduled<br>• Optionally ensuring that the preconditions of the activities are predicted to be satisfied<br>• The week plan is approved and activated to execution at the proper time |
| **Data produced** | • Week plan for the next week<br>• A new information container for the week plan<br>• An updated information model where the new week plan is activated to execution at proper time |
| **Use Case** | **6 Stakeholder coordination** |
| **Data required** | • Week plans<br>• Location breakdown structure<br>• Product-process connection<br>• Location-stakeholder connection |
| **Actions** | • The execution of activities at different locations and specified times is recorded<br>• Occupants and/or owners are notified about the activities |
| **Data produced** | • Notifications and alerts to stakeholders |

The impact of use cases is elaborated in the remainder of this section.

## 2.2 Decentralized, multi-level, and multi-version information

Consider the example of planning of a renovation project which takes place on multiple different levels and by multiple different actors (Figure 2). The owner provides the requirements for the scope, costs, and time frame of the project. The construction company creates a master plan for the project, based on quantity estimates to determine the amount of work needed, and on high-level, experience-based ordering constraints between activities. The plan is at the aggregate level and activities not directly executable yet.

At the more detailed levels, the construction company together with subcontractors and vendors creates more detailed plans and schedules. They are typically directly related to building objects of the renovated building: spaces, elements (e.g., windows, walls, pipes), zones (e.g., residential units, stairways, technical spaces) as well as systems and devices (e.g., HVAC, plumbing, etc.). Since activities are linked to different building objects (Figure 2) as well as other construction entities, much more data about the execution can be gathered – including the positions of building objects, equipment, labour, and material batches, the occupancy, temperature and humidity of the work location, and so on.

The concrete tasks are further prepared as week plans in a rolling manner, each plan specifying the guidance for execution and monitoring for one specific week. In advanced projects even daily plans may be created for closer coordination.

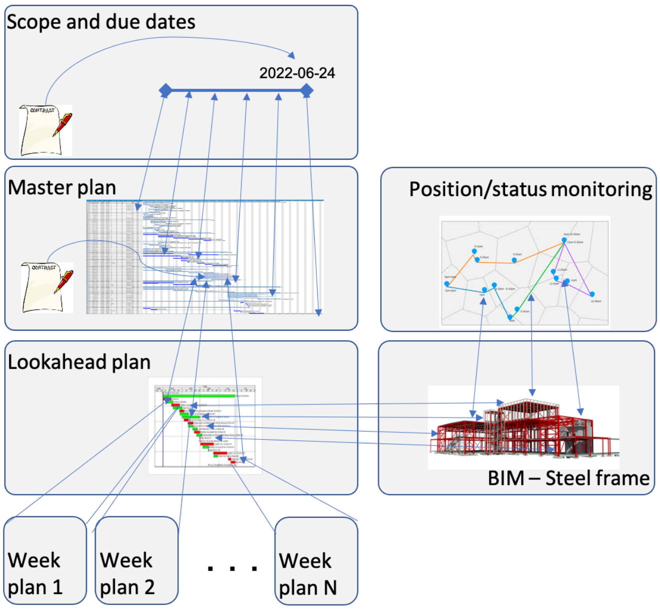


**Figure 2: Multiple levels of planning and scheduling**

The higher-level plans act as constraints on the lower-level plans, but the lower-level plans can also force changes to higher-level plans. There is a need for the information to flow to both directions. The Linked Data technology is one enabler for such communication since it can establish connections between different schedules – even if maintained in a decentralized manner – although the actual coordination functionalities need to be implemented by the level of planning and scheduling applications.

### 2.2.1 Representing the evolution of plans

As can be observed from the example, the first representational complexity in modelling the processes in the delivery phase of a renovation project stems from the following setting:

1. There are going to be *many different plans* during the project, each of which specifies the activities to be executed in the project – and many different designs that specify the goal state of the execution.
2. New plans and designs are created *when executing previously approved plans*, supplementing, or replacing the previous plans and designs, usually after separate approval activities.

Moreover, there is a range of complexities in the management of plans and designs guiding the flow of work in the delivery phase of a renovation project:

- *Decentralization*: Plans and designs are produced in a decentralised manner. That is, there are different actors that produce different designs and plans, coming from different organizations or from different levels in an organization. There are consequently needs (1) to exchange complex sets of interlinked models and plans (so-called multi-model containers), (2) to take into account the identifiers in the external systems that may be crucial for the coordination of further actions, and (3) to record the origin of the information to facilitate possible change management.
- *Multi-level*, *multi-discipline*: The plans and schedules are produced in multi-level fashion, starting from the requirement level specification by the client or owner, the masterplan by the construction company, and lookahead plans, week plans or even daily plans at the levels closer to execution. The designs are produced in different levels of detail, and in different disciplines: architecture, structural, and building systems. Plans should therefore (1) be able to refer to the higher-level plans in a proper manner, and (2) the multitude of plans should be managed in a coherent manner.

- *Multi-version*: There can be multiple versions of same plans and designs. Some of the versions may be produced according to a plan, and some are created in a reactive manner to address some issue raised at the execution time.

Those plans and designs that are *more detailed* generally supplement the previous ones, in which case the previous designs and plans would still be valid and can be used for monitoring or comparison purposes, even though the execution is guided by the more detailed designs and plans. In contrast, when a *new version* is produced of an existing plan or design, it will generally replace the previous version.

This multitude of individual design and plan datasets that is produced during the project execution needs to be managed carefully. There is a critical need to keep track of *which datasets are valid at each point of execution*. Some datasets become obsolete as they are replaced by new versions; they must not confuse the execution of the project anymore. However, there is simultaneously an important need *to retain the history of the execution* – for instance, what datasets were in effect at a given milestone of a project – to be able compare the execution to original plans and designs, or the settle disputes between parties.

Consequently, an ontology that aims to represent the flow of work in the delivery phase of a renovation project needs to be able to represent the relations of produced datasets and manage them in a manner that proper information is provided to information consumers, using the plans and designs that are in effect at each specific moment of information access.

### 2.2.2 ISO 19650 Information management using BIM

The basic structure for BIM-based information management is defined in the ISO 19650-1 and ISO 19650-2, based on concepts of *Information model* and *Information container*. The relevant concepts are shown in Table 2. The central concept is *Information container* that corresponds roughly to a dataset, carried in a file or a database. Each scenario, plan, design, or breakdown structure can be represented in an Information container. *Information model* is a set of Information containers; since it forms a hierarchical structure, an *Information model* can also include other *Information models*. Therefore, the *Information models* and *Information containers* form a similar tree structure as folders and files.

**Table 2: ISO 19650 terms for information management (ISO19650-1, 2018)**

| | |
|---|---|
| *Information model* | “A set of structured and unstructured information containers” |
| *Information container* | “A named persistent set of information retrievable from within a file, system or application storage hierarchy” |
| *Information container state* | “Each information container can be in one of the following states: work in progress, shared, published, or archived.” |
| *Project information model* | “Information model relating to the delivery phase” |
| *Asset information model* | “Information model relating to the operational phase” |
| *Federation* | “Creation of an information model from separate information containers” |
| *Level of information need* | “Framework which defines the extent and granularity of information” |
| *Common data environment* (CDE) | “Agreed source of information for any given project or asset, for collecting, managing and disseminating each information container through a managed process” |

*Project information models* and *Asset information models* are specialized *Information models* for projects and assets, respectively. *Federation* strategy means how the hierarchical structure of an *Information model* is created. *Level of information need* (LOIN) is a concept to capture the design progression based on the needs of consuming tasks. It can be managed with similar approach to LOD levels – each LOIN version of a model represented in its own container – although LOIN is not based on predefined set of levels in contrast to the most LOD frameworks preceding its definition. *Common Data Environment* (CDE) is the system or source that takes care of maintaining the *Information model* and providing the information in the included *Information containers* to the consumers of the information.

Different federation approaches for information models are proposed in the Appendix of ISO 19650-1. The suggested discipline-based and security-based federations are shown in Figure 3. The federation approach can be a combination of those presented in the figure, or something entirely different as well; the design of the strategy depends on the needs and capabilities of each project: a sensitive government project could first adopt the security approach and under it the discipline-based federation, while in ordinary projects the security-based federation may be unnecessary.

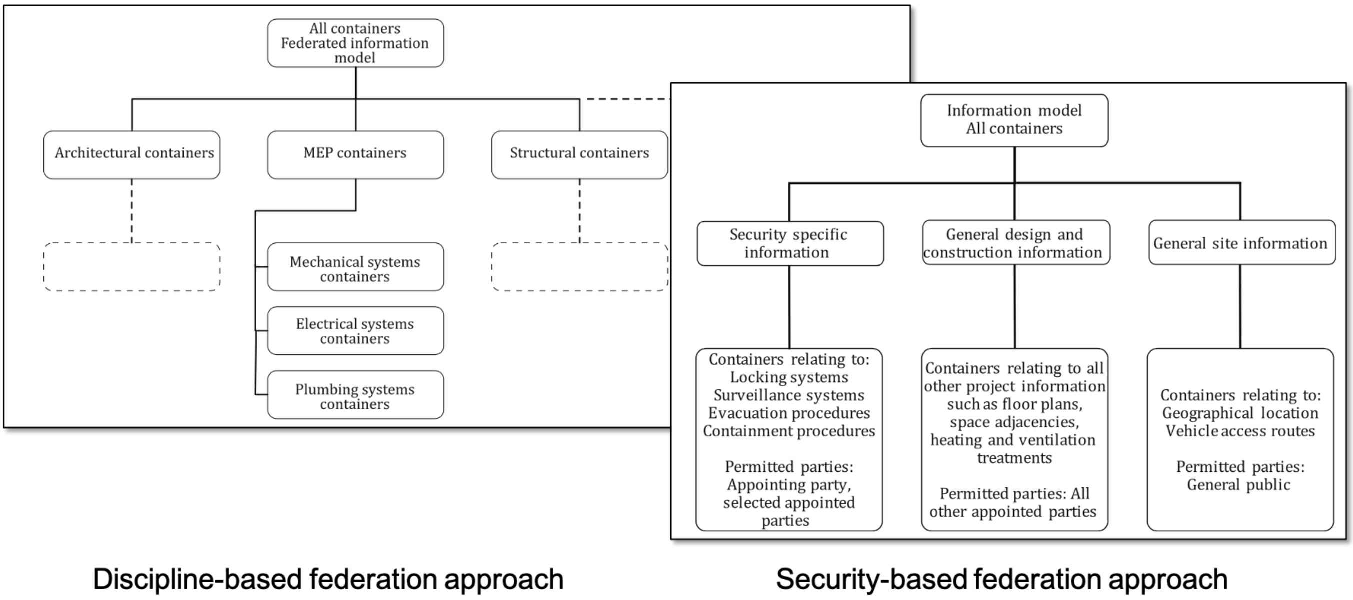


**Figure 3: Examples of possible federation approaches (ISO19650-1, 2018)**

In BIM4EEB and in Digital Construction Ontologies the concepts of ISO 19650 are interpreted in the framework of Linked Data. This means that the content of each Information container is assumed to be represented in its own RDF graph. The assumption is that the dataset is either directly produced and maintained in RDF, or that if it has been produced in another format, and there is a converter that can translate the contents into RDF. For example, BIMPlanner creates and maintains its planning data in RDF, while IFC models are produced according to IFC schema and usually exchanged in Step Physical File Format, but there are converters that can translate them into RDF compliant with the ontology version of IFC.

### 2.2.3 ISO 21597 Information container for Linked Document Delivery

The concepts of ISO 19650 Information container and Information models also bear a strong similarity to the concepts defined in ISO 21597 Information Container for Linked Document Delivery (ICDD). ICDD defines the Container ontology (ct:) that specifies the concepts of ct:ContainerDescription, ct:Document, and ct:Linkset, as shown in Figure 4.

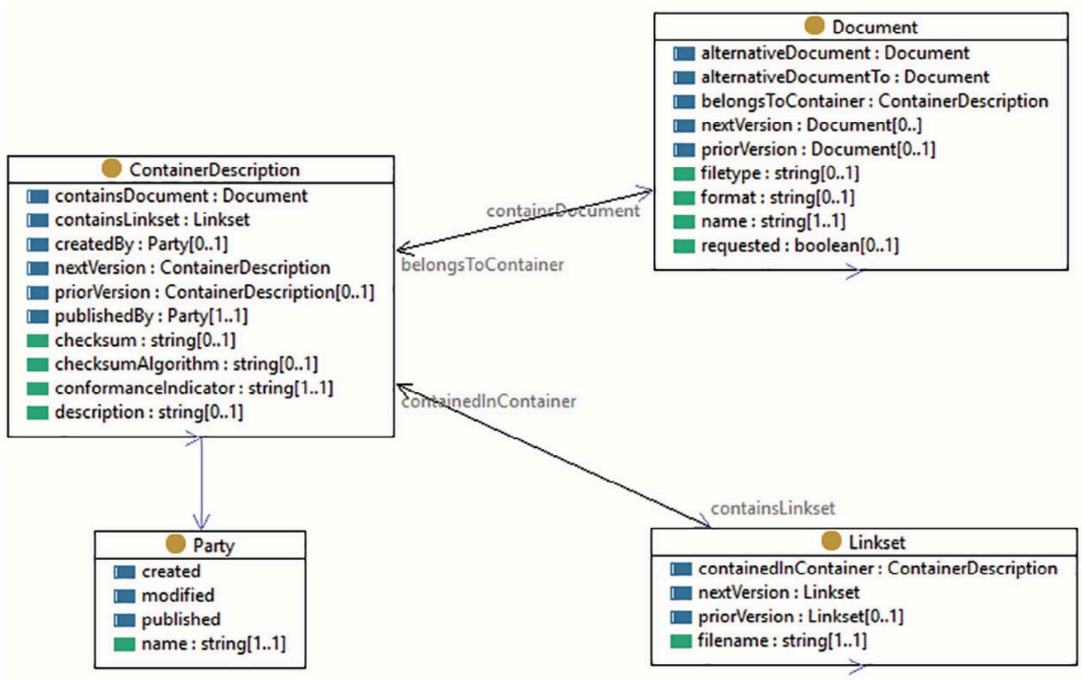


**Figure 4: ISO 21597 ICDD Container ontology**

The container concept in ICDD is different from that of ISO 19650. Indeed, the ISO 19650 Information model corresponds to the ct:ContainerDescription and the ISO 19650 Information container to the ct:Document. The ct:ContainerDescription is intended to contain multiple documents (ct:containsDocument) and linksets (ct:containsLinkset). The documents are interlinked using the links in ct:Linkset. The links in a ct:Linkset are not ordinary RDF triples, but structures that can be regarded as cross file links. Each end of a link is specified by (1) the identifier of a document and (2) internal identifier within the document.

Again, BIM4EEB and Digital Construction Ontologies take a Linked Data view to the ICDD containers. It is assumed that the contents of the documents can be converted into an RDF graph and stored in its own named graph. Moreover, it is assumed that the contents of each linkset can be converted to RDF triples, connecting the identified elements into the RDF graphs, and stored as its own named graph.

The significance of ICDD is in exchange of data between parties; the capability to exchange several interlinked documents as one package is needed, for instance, to document contractual milestones. In BIM4EEB the purpose is to digest these packages and treat their content as Linked Data.

### 2.2.4 DCAT2 Data Catalog ontology

In digitalized renovation projects there are datasets consisting of sensor data, status updates, results of questionnaires, and other similar data. The DCAT2 Data Catalog ontology is to describe these kinds of datasets, and is therefore relevant to BIM4EEB and Digital Construction Ontologies. The main structures of DCAT2 (dcat:) are shown in Figure 5.

In terms of alignment the dcat:Catalog corresponds to the ISO 19650 Information models and the dcat:Dataset to the ISO 19650 Information container.

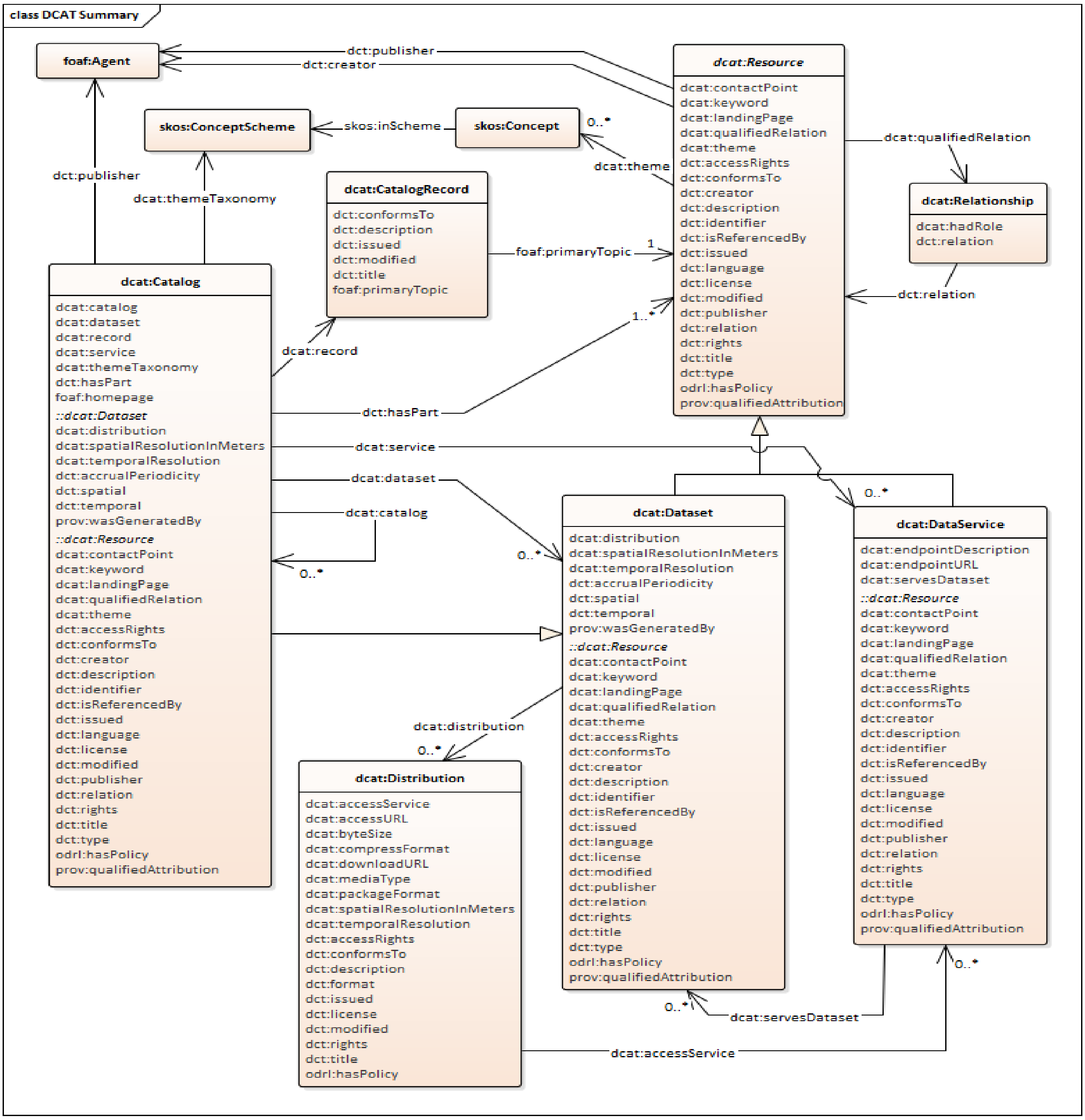


**Figure 5: DCAT2 Data Catalog ontology**

The Digital Construction Ontology has a set of concepts that bring these different models together and align with them, as presented in the later chapters.

### 2.2.5 Temporal evolution of information

Besides managing the information models and information containers that are created during the execution of a project, there is a need to support the temporal evolution of information during the project execution. This is information about things that have happened: the observations of positions of workmen, or occupancy of a work location, the progress updates about the start and finish times of activities, the creation of an information container (such as a BIM model or a master plan), an issue that has been noticed during the execution, and so on.

This temporal information is more fine-grained than the information containers included in information models. In many cases it only concerns one individual property (such as start time, temperature, position), and needs to be managed at the level of properties. In RDF it is only possible to add information to properties if they are reified or objectification. This same problem has been addressed in existing ontologies, as shown in Figure 6. The figure indicates the parallel constructs in QUDT, SSN/SOSA, and Saref Core related to the entities, properties, types of properties, activities, property states, time, value and unit. Not all ontologies support all of these aspects but figure shows, how they address them.

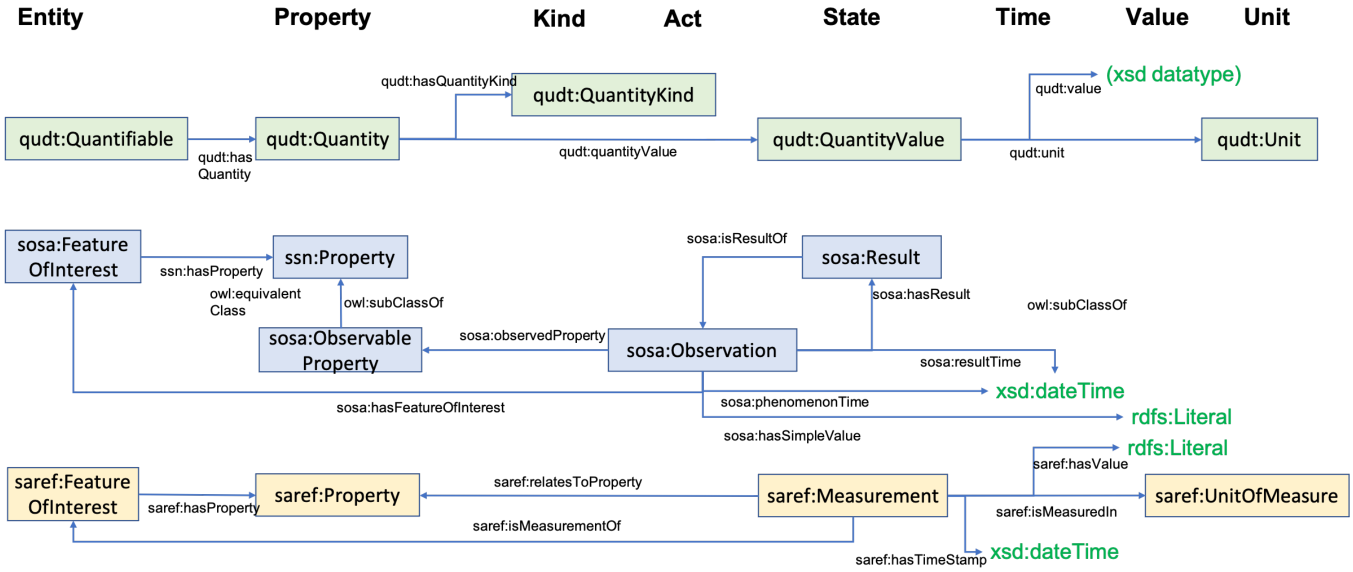


**Figure 6: Objectified properties in QUDT, SSN/SOSA and Saref Core**

There is also a similar structure in the OPM ontology. This basic construct is established and can be followed to capture the evolution of property values over time.

## 2.3 Entities and dependencies

The construction phase of a renovation project consists of sequences of activities performed to construction objects. In addition to construction objects themselves, the activities have dependencies to a range of other kinds of entities: labour crews, equipment, material batches, workspaces, information objects (e.g., BIM models, drawings, or specifications), and environmental conditions (Koskela 1999, 2000) (Garcia-Lopetz 2017).

The execution of activity requires that all its ingredients are at a correct state at the beginning of the activity, and furthermore, that some flows also remain at the proper state all the way throughout the execution. For example, a painting activity requires that the wall to be painted (the building object) is at a proper state after the previous activities (e.g., plastered), that the painters (the labour crew), ladders and brushes (equipment), and paint cans (material batches) are at the work location, and that the painting instructions (information objects) would be accessible to the painters. Moreover, the resources such as painter and the equipment should be able to remain at the work location for the whole duration of the activity.

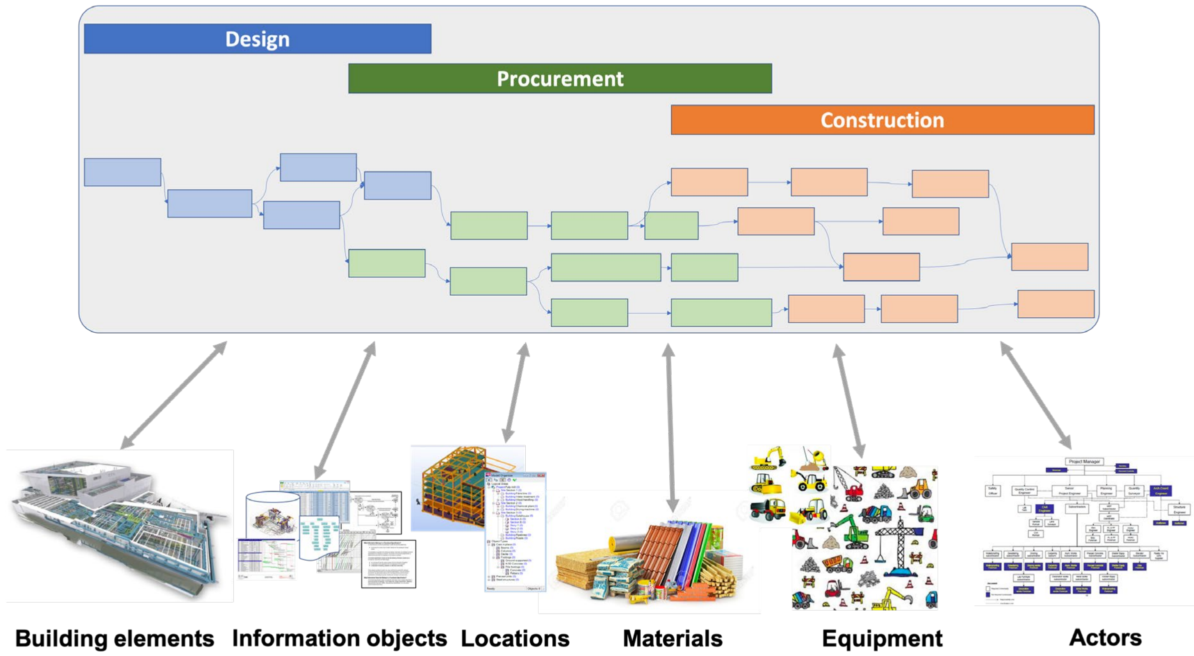


**Figure 7: Different dependencies of activities**

One of the reasons for low productivity of construction is the complexity of dependencies of activities and insufficient methods to manage them. The information about dependencies is genrally not managed in an organized way and up-to-date information about the states of the activity ingredients is missing. Consequently, construction workers spend a lot of time on looking for missing materials, equipment and information. According to recent studies where the locations of workers were monitored: they were found to spend around 20% of work time in the location where they should have been according to the construction plan (Seppänen, 2019).

The concept of a *flow* has been adapted to the construction domain from previous work on lean manufacturing (Kafcik, 1988, Womack, 1990, Koskela, 1999, 2000). In lean manufacturing, a flow consists to entities to be transformed – that is, components and materials – that move from a stationary production resource to another. The goal of production system design and production management work is to keep components in a constant movement from one task to the next, maintaining a proper order and avoiding interruptions. Factory manufacturing has a repetitive and stationary nature which makes it possible to focus on the improvement of the velocity and simplification of flows. In summary, a factory has a *static background of resources* and a *dynamic foreground of flows* consisting of components and materials.

Construction, however, is more complex and the foreground and background get mixed. Some resource-like entities are indeed stationary (such as spaces) but many others are non-stationary (labour, equipment, trucks). Likewise, while some product-like entities are non-stationary (components and materials), some others are stationary (spaces) and some transform from non-stationary to stationary during installation. Morever, as can be noticed, the spaces within a building can simultaneously be considered as resources of some activities – interpreted then as workspaces – and the products of other activities; indeed, the creation of protected spaces can be regarded as the ultimate goal of construction.

Moreover, due to the unique nature of each building and related building organization, there is much less repetition in construction and therefore the flows do not form identifiable and visible "*rivers*" but rather "*networks of small creeks*" spreading around. Even with these reservation, however, the identification and management of activity ingredients based on the approach of lean construction has turned out to be highly successful. It has led into the emergence of efficient construction management methods and practices:

the Last Planner, Loction-based Management System, and Takt-time planning.

### 2.3.1 Flows and property states

Flows related to renovation and construction activities can be classified into a small set of categories. Table 3 shows the named used by Koskela (1999, 2000), and the corresponding terms used in this work.

**Table 3: Flow categories**

| | **Koskela (1999, 2000)** | **DiCon** |
|---|---|---|
| *1* | *Precedence* | *Object* |
| *2* | *Material* | *Material batch* |
| *3* | *Labour* | *Agent* |
| *4* | *Equipment* | *Equipment* |
| *5* | *Workspace* | *Location* |
| *6* | *Information* | *Information* |
| *7* | *External* | *External* |

The categories indicate the different types of ingredients that the execution of a construction activity requires. The exact names of the categories are not important for construction management: from management perspective, instead of *categories*, it is essential to know *the set of specific flows of each activity*. An activity can have several flows belonging to one category and not necessarily any flows in some other categories. For example, precast installation requires two labour flows (a crane operator and an installation crew), while drying activities none. Also, movement activities do not need specific workspaces, and some indoor activities are not sensitive to external conditions.

The information about possible flows comes primarily from sources that define the *end product* (the building) and *resources*. Nowadays, the primary source of end product information are BIM models; component, material, workspace, and information flows can be extracted from them. The information about resource flows – labour, equipment, and workspaces – is often maintained in construction management systems.

In addition to know which flows relate to which activities, it is important to know the *state of the flow* that is required by an activity. For instance, the paints and painting equipment (flows) must be in the location (state) where the painting activity will be done before the painting can begin. The states of the flows change as the result of execution of activities. Figure 8 shows a set of flows encountered by the activity A1, together with the preconditions, execution conditions, and effects of activities. In the figure the essential flow concerns a component that is transformed by the activity and whose status changes as a result; this would belong to the category of precedence flows.

It is useful to be able to identify as many flows as possible and consequent dependencies before the execution of an activity since it allows to prevent problems to take place during the execution. However, if some flows cannot be identified or monitored, they can be still managed during the execution time by normal everyday construction management work. The flow model provides an opportunity for advanced planning and the use of sensor-based monitoring data.

### 2.3.2 Effect-precondition dependency

To execute A1 (Figure 8), some of the flows must satisfy the given preconditions. The component that is the focus of the activity – such as a wall to paint or a window to install – must be in a correct status before the execution. For instance, a wall needs to be in a status smoothed before painting can begin, or a window must be completely manufactured before installation is possible.

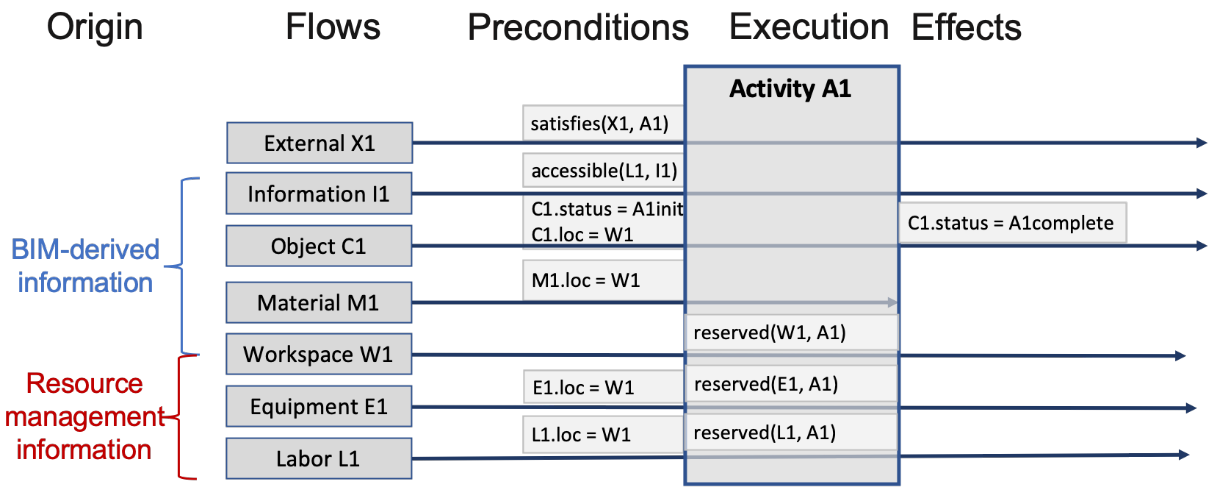


**Figure 8: Activity, flows, preconditions, and effects**

A major type of preconditions related to all physical ingredients of an activity – components, materials, equipment, and labour – is that they have to be located at the place where the activity is executed. There are additional preconditions that the relevant information (drawings, specifications, installation instructions) has to be accessible to the labour of the activity. Nowadays, as information is increasingly delivered through digital channels, physical paper prints do not need to be located at the workspace; rather, accessibility means that (1) the labour has mobile devices, (2) the specific information has been produced, and (3) the labour has access path (connectivity) and (4) access rights to the information. Information itself can be in a BIM collaboration system, a document management system, or a messaging system. Finally, some activities have preconditions that refer to external, environmental factors such as proper temperature or humidity levels.

The counterpart of preconditions are the effects of an activity. They come in two variants:

- *Add effects*: what new conditions will be in effect after the activity;
- *Remove effects*: what preconditions do not hold anymore after an activity.

In Figure 8Figure 8, there is one add effect: the change of the transformation status of the component C1. Activity A1 is thus a way to transform the component C1 from status A1init to A1complete. All value adding activities can be represented as such transformations. For example:

- painting is an activity to transform a wall from status plastered to status painted;
- transportation transforms the location of a shipment from origin to target;
- installation of a component transforms object from uninstalled to installed.

The chaining of transformations creates sequences of activities. For instance, consider a component w1 of the type wall. Since there are activities:

- panelling: precondition is the status frameConstructed, effect is the status panelled,
- plastering: precondition is the status panelled, effect is to add the status plastered,
- painting: precondition is the status plastered, effect is to add the status painted, and
- skirting: precondition is the status painted, effect is to add the status skirted.

When these preconditions and effects of different activities are considered and matched with each other, the only resulting workflow of the activities is

- paneling - plastering - painting - skirting

Thus, when one activity has an effect that another one requires, it creates a dependency between the activities. Sometimes, as in the example above, these relations result in a fixed sequence of activities but often they just create constraints on the set of possible sequences.

### 2.3.3 Competing resource needs

Certain ingredients of activities – labor, equipment and workspace – need to be managed as *resources*. Resource is a role concept: it is not an inherent characteristic of any entity to be a resource. It is rather a role that an entity can play with respect to an activity. For example, to provide protected space – that is, a location – is the primary product of building construction; however, during construction, locations also play the role of a resource with respect to activities, especially in indoor construction activities.

Since many different activities may need to reserve the same type of resources – such as all painting activities require painters, or all activities at the same wall (paneling, plastering, painting, skirting) need overlapping workspaces – the available resource capacity limits the number of such activities that can be in execution simultaneously. This establishes temporal constraints between activities. If the resource capacity is one, the result is disjunctive constraints between each pair of activities: each activity needs to be executed either before or after every other activity requiring the same resource. Otherwise (when resource capacity is more than one), the available capacity poses the upper limit to concurrent activities. For example, if there are three painters available, at most three activities that each require one painter can be in execution simultaneously.

If temporal constraints are not identified, they need to be handled during on-site construction management activities. There will be conflicts if more activities want to use a particular type of resource than what is its available capacity. That kind of conflicts cannot be completely avoided, but their frequency can be reduced and they can be anticipated earlier if different resource requirements are properly modeled and managed.

### 2.3.4 Plans, variables, and constraints

The planning can be either specified in terms of goals to achieve (like in routing a delivery to goal location), or in terms of a larger activity to execute (like constructing a building). The objective of planning is to determine the necessary executable activities. In larger efforts, plans are usually developed at different hierarchical levels; in construction projects there can be a masterplan, phase plans, lookahead plans, week plans and even daily plans. At each level the plans of the previous level are refined to produce more limited, concrete and executable activities, more clear and unambiguous constraints between the activities, and/or more exact resources assignments and schedules.

There is a large variety of methods and approaches for plan generation. The methods used in practice are mostly manual and based on previous plans, plan templates and/or experience. Recently also automatic plan generation methods have emerged, especially within the AI planning research. During the coming years, planning will likely develop into increasingly computer-supported task, where a human planner evaluates and selects from possible plans or plan fragments to create the finally approved result. There is thus a need to combine the expertise of a human planner with the algorithmic power of automatic planning.

*Mixed-initiative planning* is an approach that combines manual and automated planning approaches by allowing people and algorithms to collaborate in a planning process through a *shared plan representation* (Tate, 1998, 2001). A shared representation and support for mixed-initiative planning can also pave the way for the development of more collaborative planning practices across different hierarchical planning levels, including between managers and workmen.

Since the execution of a plan often breaks some of the assumptions that were adopted during the plan generation, most plans end up being incomplete; they may contain various kinds of unresolved issues such as conflicts, open goals, non-specific actions, and so on. Therefore, any realistic plan representation must be able to represent also *incomplete plans*.

In the <I-N-OVA> framework (Tate, 1996a, 1996b) this is achieved by representing plans composed of following ingredients:

- *Issues*: The open goals, inexecutable tasks, or broken constraints in a plan – all signifying the aspects of plans that need to be fixed. The issues can be considered as implied constraints on the plan; the resolution of issues happens through the introduction of actual constraints to the plan.
- *Nodes*: The activities contained in a plan. Each activity hosts several variables for different flows, resource assignments and time windows.
- *Constraints*: Constraints for activity ordering, variable values (co-designation and non-co-designation constraints), or for auxiliary constraints specific for activity-type.

The representation of the resource requirements can be implemented through the capability constraints, the activity ordering is specified with temporal constraints (Cox, 2017), and the representation of preconditions and effects of activities are specified with state constraints adopted from (Fox, 2002). Constraint representation allows the use of constraint propagation algorithms to identify constraint violations. The violations can be further recorded in plans as open issues. The set of issues in a plan form a sort of to-do list for completing a plan (Tate, 1996a). The issues in the to-do list can be picked for resolution by a constraint handler that can be either a human or a machine.

A resolution of an issue can generate derived issues, create additional activities in a plan (with associated variables), and pose new constraints. When all open issues are resolved, the plan is completed and ready for execution. If issues cannot be resolved, some constraints may need to be relaxed, for instance, by increasing resources or adjusting deadlines. Overall, planning can be regarded as collaborative resolution of the issues in a plan. The process can span a period of time and take place within a group of human and machine agents, since there is a proper shared representation of the incomplete plan throughout the process.

### 2.3.5 Agents, appointments, and teams

An importance concept in management of the flow of work in any process is the concept of *agent* and agency. The activities are time-limited intentional processes, and the intention is provided by an agent. The agency needs to be modelled as a relation between an activity and an agent.

Well known ontologies that represent agents are FOAF (Brickley, 2014b), Org (Reynolds, 2014), PROV-O (Lebo, 2013) and FIBO (Bennet, 2013). Figure 9 shows the relevant concepts in these ontologies with some suggested alignments with dashed lines.

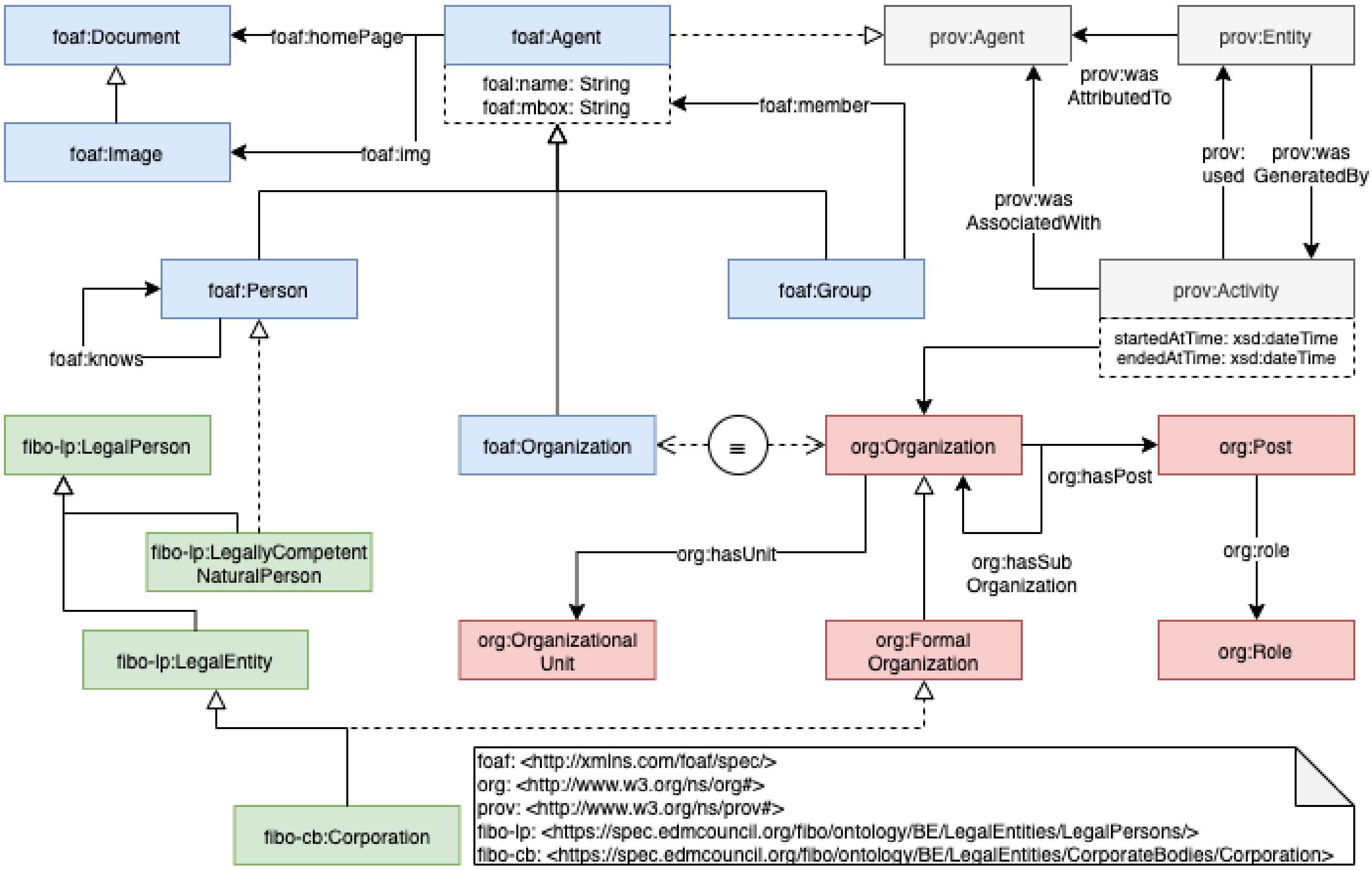


**Figure 9: Agent concepts in existing ontologies**

FOAF is aimed at to be an ontology for social networks, formed by the foaf:knows relation between people (foaf:person). It has also been used as an element in the WebID authentication mechanism. Org ontology specifically tries to capture the internal structures of organizations: posts, roles, suborganizations, and so on. FIBO is a financial industry business ontology; a broad collection of interlinked ontologies. It is useful to capture the legal status of different agents, which is needed to define, for instance, the concept of an owner. PROV-O is about provenance data, and the concept of agent is central in it. However, in PROV-O agents are also assumed to include computational agents that as of now do not have any legal status or capability to carry responsibility. The prov:Agent can be considered to be more generic concept than foaf:Agent.

One key area of ISO 19650 is to address the information exchanges between different parties in BIM-based projects. Its concepts for agents and resources are summarized in the Table 4.

**Table 4: ISO 19650 concepts for agents and resources (ISO1650-1, 2018)**

| | |
|---|---|
| *Actor* | "Person, organization or organizational unit involved in a construction process"<br>Note1: "Organizational units include, among others, departments and teams"<br>Note2: "[…] construction processes take place both during delivery phase and operational phase." |
| *Appointment* | "Contract, professional services agreement."<br>"Agreed instruction for the provision of information" |

| | |
|---|---|
| | "This term is used whether or not there is a formal appointment between the parties" |
| *Appointing party* | "Client, asset owner/operator" |
| *Appointed party* | "Supplier "<br>"Provider of information concerning works, good or services" |
| *Lead appointed party* | E.g., a general contractor |
| *Delivery team* | "Lead appointed party and their appointed parties"<br>"Note 1: A delivery team can be any size, from one person carrying out all the necessary functions through to complex, multi-layered task teams. The size and structure of each delivery team are in response to the scale and complexity of the asset management or project delivery activities."<br>"Note 2: Multiple delivery teams can be appointed simultaneously and/or sequentially in connection with a single asset or project, in response to the scale and complexity of the asset management or project delivery activities."<br>Note 3: A delivery team can consist of multiple task teams from within the lead appointed party's organization and any appointed parties." |
| *Task team* | "Individuals assembled to perform a specific task" |
| *Capability* | "Measure of ability to perform and function" |
| *Capacity* | "Resources available to perform and function" |

The ISO 19650 concepts concern the groupings of agents relevant to the delivery phase activities, their organizational relations within a project, and capabilities and capacities as resources.

## 2.4 Current state, measures, plans, and execution

To map the information contents related to the flow of work that need to be captured in the Digital Construction Ontologies, the BIM4EEB toolkit is considered as a motivating scenario. Figure 10 shows the setup of different tools, from the perspective of how to manage the flow of work in the delivery phase. The figure indicates the following relevant stages:

- Current state, in which it is important to gather information about the building (using the Fast Mapping tool developed in WP5) and about the occupants profiles and preferences (using the BIM4Occupants tool developed in WP7)
- Renovation scenario definition stage, in which the renovation measures to be implemented in the renovation project are analyzed and selected (using BIMEaser developed in WP6 for energy simulations and definition of target criteria to meet in the project – the owner's project requirements), and technically detailed (using AUTERAS and BIMcpd developed in WP6).
- Renovation planning, in which the renovation measures identified in the previous stage are developed into an evolving plan of work for the renovation project, at multiple successive levels and also extending over the period of execution. The planning is done with the BIMPlanner tool developed in WP7.
- Renovation execution, in which the plans are executed, and rolling detailed plans for the next period (such as week or day) will be created. The execution needs to be monitored as well. BIMPlanner (WP7) is based on creation of week plans based on the higher-level master plan and the spatial decomposition of a building.

The development of plans for the delivery phase requires as input the BIM model of the building, the selected renovation scenario, and external constraints to the project. The renovation scenario determines the renovation measures are activities. Many of the activities have a repetitive character. For instance, façade insulation is installed to each section of the façade, each of the windows is replaced with a new one, a particular multi-phase process needs to be executed in each space, and so on. The specific activity instance needs to be created for each repetitive element and scheduled respecting all applicable precedence and resource constraints.

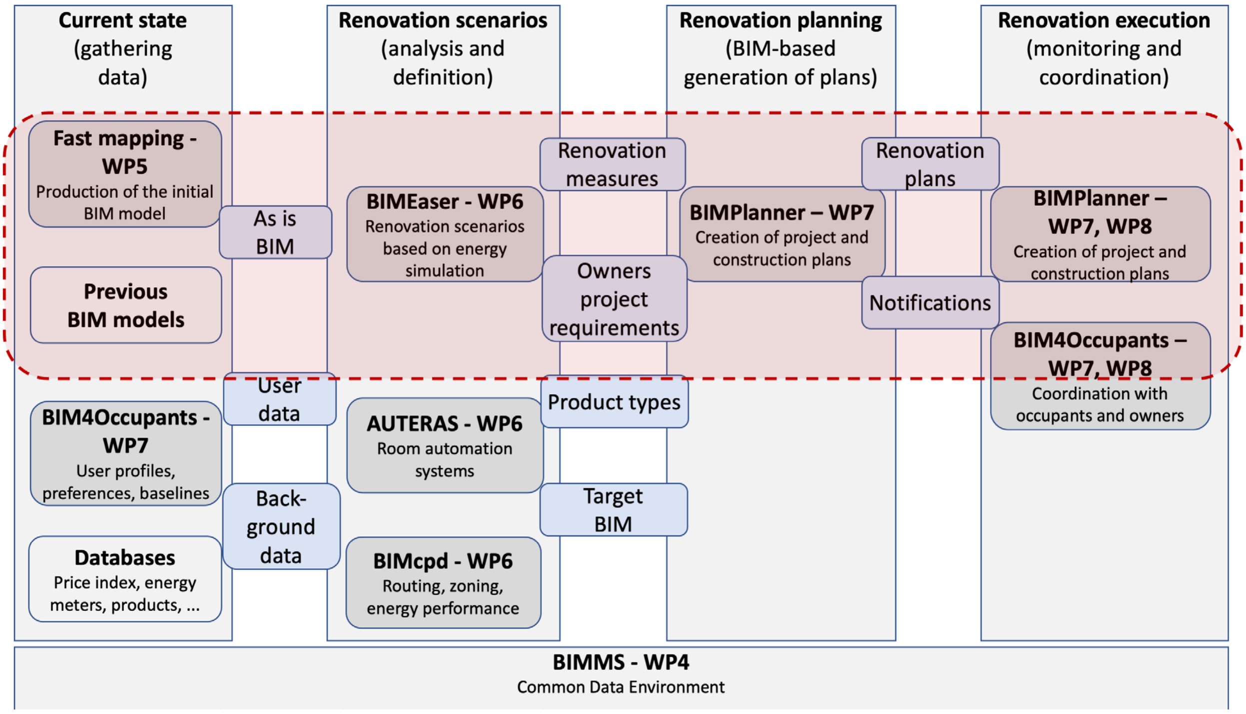


**Figure 10: Renovation tools at different stages**

The elements for which the renovation measures are repeated are derived from the BIM model, but usually the development of the definite set needs additional input. In the interest of efficient construction management, it is typically necessary to develop a specific location breakdown structure (LBS) that contains those work locations that are relevant to the project execution. The locations in an LBS are not directly the spaces that are available in a BIM model. A location can be an aggregation of multiple spaces, a part of one space, or even something that has no corresponding space in the BIM model, like external areas at the construction site.

BIMPlanner is a location-based planning system which means that the renovation measures expressed as higher-level activities in a master plan are decomposed and instantiated as location-specific activities. Their execution times are defined and based on that week plans can be created. The execution of week plans are monitored by recording the actual executions times.

The location-based schedule of activities forms the basis for notifications to the occupants about the disturbance caused by the work, as carried out by BIM4Occupants.

## 2.5 Other

### 2.5.1 Ubiquitous changes

There are many different types of changes that construction processes encounter during their execution:

- *Operational changes*: Due to their numerous dependencies, activities have many different ways to fail. If any one of the dependencies is not satisfied, the execution of the activity is usually blocked, or the execution will lead to lower quality of work which increases the probability of eventual rework. The planned workflows face constant pressure from operational changes: delays, unexpected interferences between activities, and so on. If plans were to provide any operation guidance for construction work, they should be revised or reconstructed frequently.
- *Planned refinement of information*: The second dimension of changes concerns the evolution of information during the execution. There are many different variations of that. First of all, the design process has been planned to go through a set of previously envisioned refinements of information. The designs are developed through a sequence of levels of detail, from conceptual designs to designs that takes the details of construction solutions into account. Similarly, the project and construction plans are gradually refined from master plans to weekly plans. At the lowest levels the plans are frequently revised.
- *Unplanned downstream changes to information*: Changes in requirements or other constraints of the project can cause revisions to designs or plans that need to be adopted by downstream activities. These kinds of changes often ripple down the workflow as notifications.
- *Unexpected upstream change requests*: Designs or plans may need to be revised because of conflicts or problems encountered in the execution time. This will usually result in change requests upstream of the workflow.

What this means is that same objects or activities will have different internal structures, external relations, and property values at different stages of the project. The ontology should allow the representation of the evolution or refinement of information since there is a great interest to compare the values at different stages to detect deviations and possibly to learn from them.

### 2.5.2 Observations

The execution of construction work is typically characterized by the uncertainty about the situation: which activities have been completed, is the workspace already vacant, where are the equipment, materials or documents, and so on. Different actors can have a different understanding of the situation leading to conflicting decisions.

The traditional progress monitoring approaches – periodic collection or reporting of progress estimates – only provides rough picture of the situation. The information is often incomplete, sometimes biased, and received only after a time lag.

With the ongoing digitalization the situation on construction sites is now changing rapidly. There are new mobile tools – applications working on standard smart phones – that allow the workers to report important progress information in real time: when an activity is started or completed or when a status of an object is changed (such as a delivery is received). Furthermore, the tools can allow the immediate reporting of different kinds of issues, ranging from roadblocks to work (such as location is reserved, garbage needs to be cleaned up before the task can start, equipment is broken, and so on) to quality problems in the materials or completed work.

Moreover, sensors are increasingly used to gather information about site conditions – temperature or humidity – and about positions of actors, products, equipment, and material kits. The monitoring data helps to determine whether the execution of an activity is enabled, that is, are all the preconditions satisfied.

Imaging data – generated from laser scanning or photogrammetry – can be mapped to design models to follow the progress of construction work. The process is complex and requires understanding of not just how the building to be constructed should look at each moment of time (what building objects should be visible for the camera), but also how various process objects (large equipment, scaffoldings, formwork, material kits, and so on) look like and when they are likely to be present at the site.

### 2.5.3 Automatization

Due to ongoing digitalization, proliferation of mobile devices, better observational capabilities, and better organized information systems, there are more and more opportunities for automatization of various aspects of renovation and construction workflow management.

This includes the monitoring of the status and positions of different flows: buildings objects, resources, locations, and environmental conditions. The information for this can be received from information. Systems for procurement, construction management, and supply-chain management as well as sensor systems and mobile devices. Such data facilitates advanced functionalities for project supervision: the preconditions of activities can be checked automatically, and problems can be predicted early on.

More elaborate construction plans – that connect domain entities with activities – can be supported by automatic constraint checking functionalities, for which there are well-understood and proven implementation techniques available. Moreover, the plans can be simulated to uncover problems due to unforeseen interferences. Furthermore, the plan generation can be gradually automated using configurable and parametrized activity and process libraries and applying the algorithmic techniques studied in the field of AI Planning (Ghallab, 2016). There are a range of methods available, and although the computational complexity of widely studied algorithms has been a problem, there are also more practical variations of the techniques that have been used in real-world planning problems.

A key technology to support automatization are ontologies that help to integrate information from multiple sources and thus enable incremental adoption of more automatization.

# 3 Ontology requirements

The requirements for the ontology are divided into functional requirements that define the scope of the ontology and the required representational capabilities, and non-functional requirements that define how it should provide those representational capabilities.

**Scope**. The scope of the ontologies is the representation of *the flow of work in the delivery phase of a renovation project*. The representation is divided into modules: Contexts, Variables, Entities, Processes, Agent, and Information. The ontologies should cover the *aspects of digitalized renovation processes* that produce and consume BIM models, plans in digital format, sensor observations, as well as other systems for enterprise and project management.

**Competency questions**. The functional requirements are specified as competency questions for each of the ontology modules, shown in Table 5.

**Table 5: Competency questions**

| | | |
|---|---|---|
| **Contexts** | **dicc-cq1** | *How to store and manage datasets separately? (for versions or alternatives)* |
| | **dicc-cq2** | *Is the given statement true in the given context?* |
| | **dicc-cq3** | *What statements hold (are true) in the given context?* |
| | **dicc-cq4** | *What is the difference between a dataset in two different contexts?(e.g., between versions)* |
| **Variables** | **dicv-cq1** | *What are all the values of a property of an entity over time? (evolution of the value)* |
| | **dicv-cq2** | *What is the quantity kind and unit of a quantitative property?* |
| | **dicv-cq3** | *What are the constraints between properties? (e.g., less than, equal)* |
| **Entities** | **dice-cq1** | *What entity has a given identifier in the given scope? (e.g., the room number in a building)* |
| | **dice-cq2** | *What entities are classified in a given category in the given scope? (e.g., EG000819 in ETIM)* |
| | **dice-cq3** | *What identifiers/categories does an entity have and in which scopes?* |
| | **dice-cq4** | *What type and instance have been assigned to an entity? (prescriptive, as in product selection)* |
| | **dice-cq5** | *What entities are positioned in the given location?* |
| | **dice-cq6** | *What are the parts of a building object?* |
| **Processes** | **dicp-cq1** | *What subactivities (or leaf-level subactivities) does the activity has?* |
| | **dicp-cq2** | *What entities (or input/output entities) is the given activity acting on?* |
| | **dicp-cq3** | *In what location (or initial/final location) is the activity taking place?* |
| | **dicp-cq4** | *What equipment are needed in the execution of an activity?* |
| | **dicp-cq5** | *What resources have been assigned for the execution of an activity?* |
| | **dicp-cq6** | *What is the time when the activity is executed? (planned and actual times)* |
| **Agents** | **dica-cq1** | *Who is the agent of an activity? (a person or organization)* |
| | **dica-cq2** | *What is the consortium of the given renovation project?* |
| | **dica-cq3** | *Who is leading the consortium of the given renovation project?* |
| | **dica-cq4** | *Who are the stakeholders (owners and occupants) related to an activity?* |
| **Information** | **dici-cq1** | *What is the information model of the given renovation project? (according to ISO 19650)* |
| | **dici-cq2** | *What information containers are active in the current state? (contain current information)* |
| | **dici-cq3** | *What information (or output information) does the given activity act on? (information flow)* |
| | **dici-cq4** | *Who produced the specific information content and when? (metadata)* |

The relation of the competency questions to the use cases and information contents to be managed in renovation project is shown in Table 6.

**Table 6: Information contents, use cases and competency questions**

| Required concepts | Use cases | Relevant external ontologies/models | CQs |
|---|---|---|---|
| Temporal information about activities (start, end, duration) | 1. Master planning<br>3. Progress coordination<br>5. Week planning | OWL-Time | Dicp-cp6 |
| Work breakdown structure (activity hierarchy) | 1. Master planning<br>5. Week planning | BFO<br>ifcOWL | Dicp-cq1 |
| Types of activities in classification systems | 1. Master planning<br>5. Week planning | Classification systems | Dice-cq2<br>Dice-cq3 |
| Constraints between activities | 1. Master planning<br>5. Week planning | OWL-Time<br>CPM | Dicv-cp3 |
| Preconditions of activities | 3. Progress coordination | PDDL | Dicv-cq3 |
| Costs of activities<br>(activity costs, unit costs) | 1. Master planning<br>3. Progress coordination | ifcOWL | |
| Product breakdown structure: components of the building, subcomponent relations | 2. Task planning | ifcOWL, BOT BFO<br>RealEstateCore | Dicp-cq5 |
| Product-process connection: activity vs entities | 2. Task planning | | Dicp-cq2<br>Dici-cq3 |
| Product data refinement (assigned type and instance) | 2. Task planning<br>3. Progress coordination | | Dice-cq4 |
| Location-breakdown structure: location, sublocation | 5. Week planning<br>6. Stakeholder coordination | ifcOWL<br>BOT<br>RealEstateCore | Dice-cq5 |
| Resources of activities: requirements, assignments | 5. Week planning | ISO 19650 | Dicp-cq4<br>Dicp-cq5 |
| Management of information about the goals and plans | 4. Plan revisioning | ISO 19650<br>ISO 21597 DCAT2 | Dicc-cq1<br>Dicc-cq2<br>Dicc-cq3<br>Dici-cq1<br>Dici-cq2 |
| BIM change management | 4. Plan revisioning | ISO 19650 | Dicc-cq4 |
| Parties involved in execution of activities: person, team, company, occupant, owner | 1. Master planning<br>5. Week planning<br>6. Stakeholder coordination | FOAF<br>Org<br>FIBO<br>ISO 19650 | Dica-cq1<br>Dica-cp2<br>Dica-cp3<br>Dica-cq4 |
| Equipment needed by activity | 5. Week planning | ISO 19650 | Dicp-cq4 |
| Sensor data to activities: external conditions, positions, occupancy, representation of observed values and units | 3. Progress coordination | SSN<br>Saref<br>QUDT wgs84_pos | Dicv-cq1<br>Dicv-cq2 |
| Capturing the origin of information (who, when) | 4. Plan revisioning | PROV-O | Dici-cq4 |
| Decentralized and hierarchical planning (who, when, what plan contents) | 1. Master planning<br>2. Task planning<br>4. Plan revisioning<br>5. Week planning | PROV-O<br>ISO 19650<br>ISO 21597 | Dice-cq1<br>Dici-cq1 |

**Non-functional requirements**. The following external non-functional requirements for the ontology have been presented:

**Table 7: Non-functional requirements**

| | |
|---|---|
| **dicon-nf1** | ***Support for Linked Data****: The ontology must enable decentralized publication and utilization of renovation data as Linked Data.* |
| **dicon-nf2** | ***Modularization****: The ontology must be modularized according to the vertical and horizontal segmentation approach presented in Haller (2017).* |
| **dicon-nf3** | ***Standards compliance****: The ontology must be compliant with relevant standards:ISO 16739 IFC, ISO 19650 BIM/IM, ISO 21597 ICDD and ISO/IEC 21383 BFO.* |
| **dicon-nf4** | ***Reuse of established ontologies:*** *The ontology should reuse relevant established ontologies: BFO (Fundamental categories), OWL-Time (Temporal phenomena), QUDT (Quantities and units), FOAF and Org (Actors and organizations), PROV-O (Provenance of data), ifcOWL (BIM models), SSN/SOSA (Sensor data), Saref Core (Sensor data).* |
| **dicon-nf5** | ***Compliance with existing classification system:*** *The ontologies should support the use of existing classification systems, not repeat their contents.* |
| **dicon-nf6** | ***Open publication:*** *The ontology must be published with an open license.* |

**Non-requirements**. Information security concerns – authentication, authorization and access control (Oraskari, 2016) – are taken into account as required by the applications developed in the BIM4EEB, but their modelling at the ontologies is out of the scope in the project.

# 4 Ontology definition

This section describes the concepts of the ontology. It starts with the architecture that defines the ontology modules and their relations. Then it gives an overview of the classes in different ontology modules, after which is discusses different representational topics in more detail.

One starting point for defining the workflow ontology was the ontology work carried out in the DiCtion project that already developed a modularized structure, albeit quite different from the current one in DiCon. The alignment of DiCon with DiCtion ontologies is presented in the Deliverable 3.6 (BIM4EEB-D36, 2022).

## 4.1 Architecture

As described in Deliverable 3.1 (BIM4EEB-D31, 2021), the ontology is modularized according to the vertical and horizontal segmentation approach presented in the Semantic Sensor Network Ontology (SSN, 2021). In this approach, ontology modules do not import, even indirectly, any external ontologies, nor refer to them. All references to external classes and properties are made explicit in separate alignment modules[3]. Consequently, only the alignment modules will import external ontologies.

### 4.1.1 Modules and their relations

The modules of the DiCon ontologies described in this work are shown in Figure 11. The blue arrows indicate the vertical import relations; it is important to pay attention to the direction of the arrow. The red arrows indicate a horizontal import relation.

The lower part of the figure shows the DiCon modules in blue colour. Above that there are the alignment modules and finally the external ontologies are shown at the top. Each alignment module imports both some module from DiCon and an external module, and it ties them together with a set of alignment axioms.

The following six ontology modules have been developed in the Task 3.4:

1. *Contexts*: The division of information into different named graphs.
2. *Variables*: Objectified properties, property states, sensor observations, constraints, and values.
3. *Entities*: The fundamental categories/relations: assets, physical objects, locations and space-time.
4. *Processes*: The activity, service, behavior process, resource role, capabilities.
5. *Agents*: The agents, persons, organizations, teams, appointments, owners.
6. *Information*: Information models, containers, BIM models, master plans, week plans.

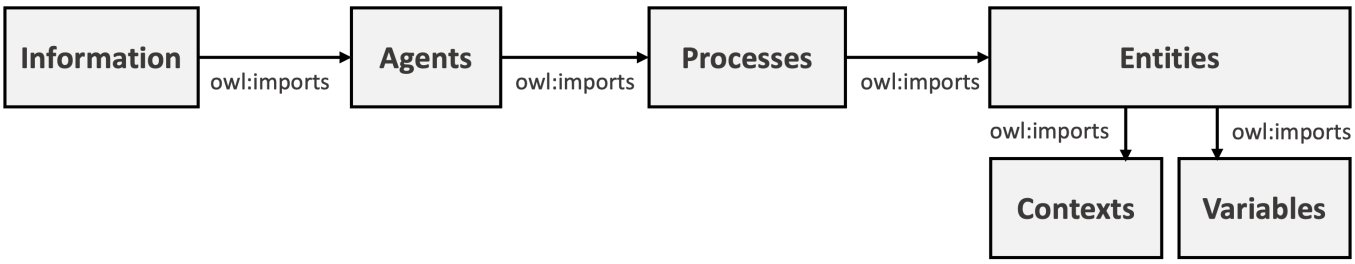


**Figure 11: Ontology modules defined in Task 3.4**

### 4.1.2 Overview of concepts

The overview of the class structure of the workflow ontologies is shown in Figure 20. It includes the

[3] This does not apply to externally defined instances that can still be referred to without alignment. Examples are units and quantitykinds defined in QUDT vocabularies, or entity instances described in DBPedia.

important classes from four of the above-mentioned ontology modules: Entities (in blue), Processes (in purple), Agents (in green), Information (in light grey), and Variables (in yellow).

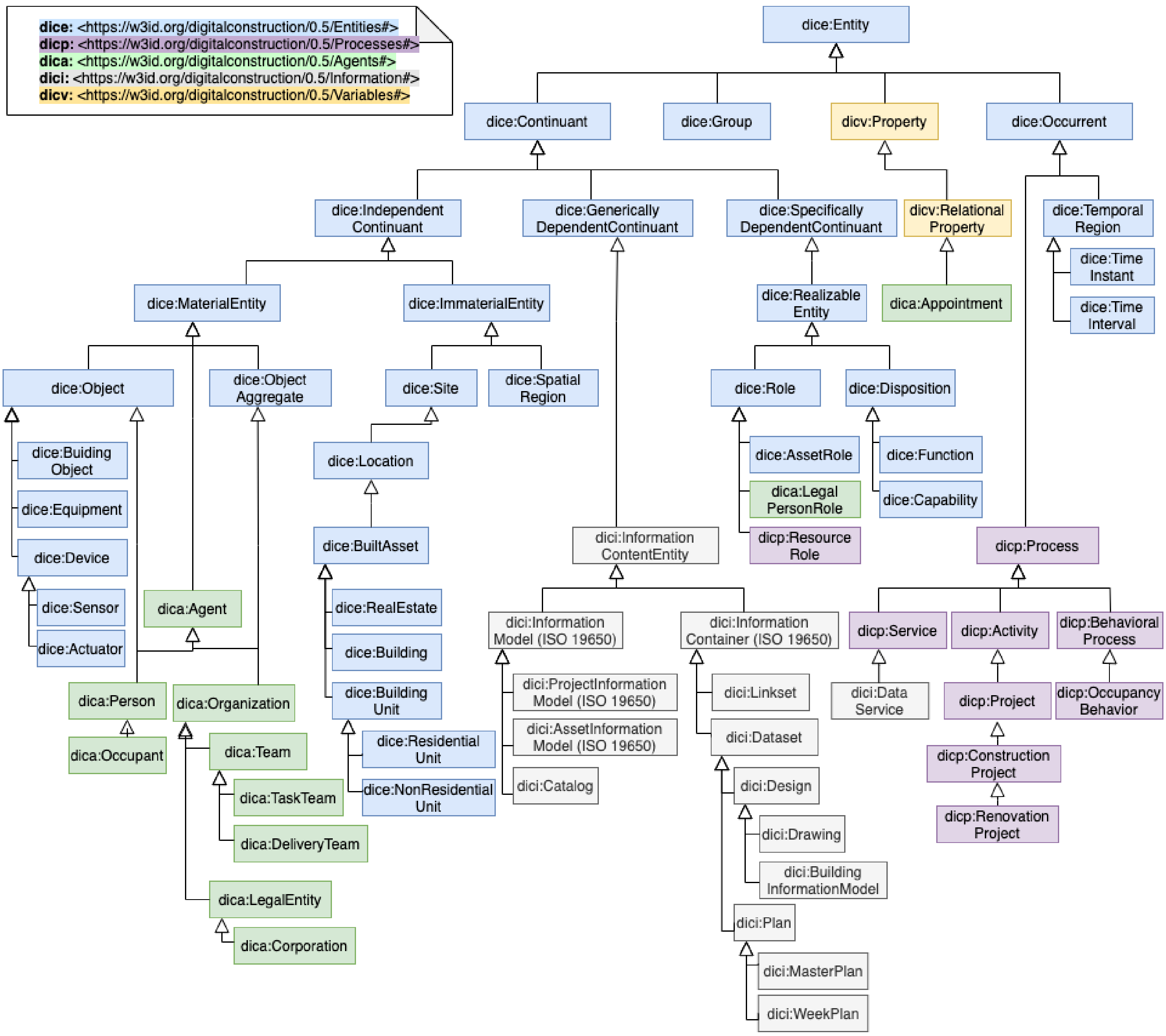


**Figure 12: The overview of DiCon workflow ontologies after classification**

## 4.2 Entities

The upper structure of the ontology is defined in the Entities module; it is based on BFO and aligned with it. Entities module also extends BFO with concepts that are specific to construction and renovation domain. For instance, it defines an asset role (dice:AssetRole) and using it the concept of a built asset (dice:BuiltAsset) to comply with the text of ISO 19650. The subclasses of built asset are real estate (dice:RealEstate), building (dice:Building) and building unit (dice:BuildingUnit). These concepts are defined as spatial entities to enable the alignment with IFC and BOT. There are also physical entities such as dice:BuildingObject, dice:Equipment, dice:Device, and so on.

### 4.2.1 Labelings: categories and identifiers

The root class *Entity* represents all identifiable entities in renovation projects, including physical objects, people, companies, models, activities and information content entities. An important goal of DiCon ontologies is to work with existing classification systems and not duplicate their contents. Another goal is to enable connections to many different external systems where different identifiers of the entities can be used. For these purposes a generic representational mechanism for labeling the objects has been included in the Entities ontology, as shown in Figure 13. There are two types of labels: categories and identifiers.

- Identifier (dice:Identifier) – An entity can have multiple different identifiers (dice:isIdentifiedBy) based on different identifier systems. They are instrumental in data linking and interoperability. An identifier can be globally unique – such as GUIDs in IFC or URIs in Linked Data – or unique in some local context – such as room numbers of a building or control point numbers in a building automation system.
- Category (dice:Category) – Construction industry and related product industries have existing and established classification systems. An entity can have an association to multiple different categories (dice:isClassifiedBy) from selected classiciation systems such as Omniclass, Uniclass, Uniformat, freeClass, Talo84, Talo2000, or ETIM. The class identifier in each of these classifications can help to access other related information such as specifications, guidelines, or standards.

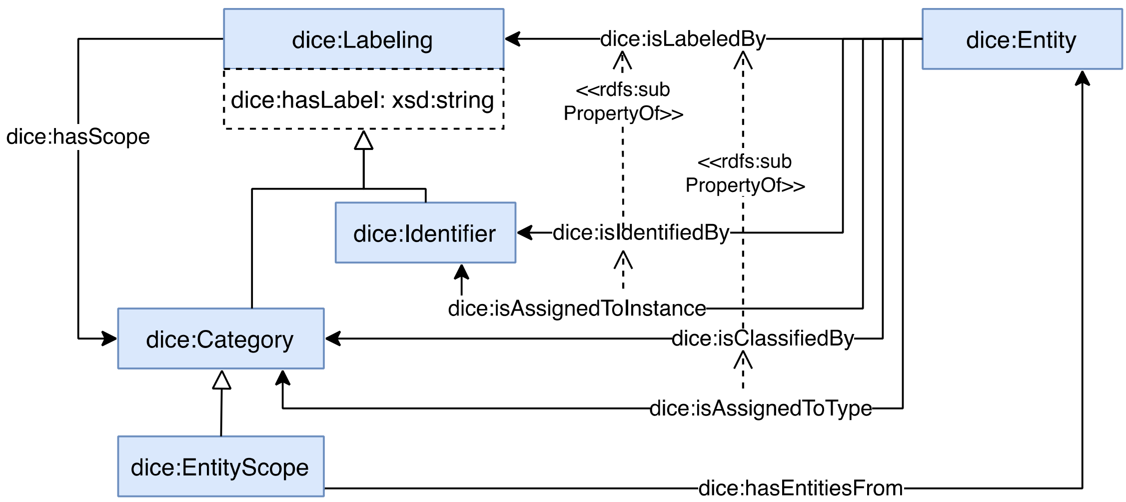


**Figure 13: Labeling of entities: categories and identifiers**

The modelling idea of multiple identifiers has been borrowed from the gist ontology (Gist, 2021).

The labelling system can also be used to represent product assignments, using the subproperties dice:isAssignedToType and dice:isAssignedToIdentifier, as explained in more detail in the Deliverable 7.3 (BIM4EEB-D73, 2021).

## 4.3 Variables

Since renovation projects are carried over time, the values of most properties are also subject to change. To capture the possibility of changes, all properties can be objectified and associated with different values at different times. The objectified relations are represented according to Figure 14. Any object (dicv:Subject) can be associated with objectified properties with the object property dicv:hasProperty. An objectified property is an instance from a class dicv:Property. The property object can be associated with property states from class dicv:PropertyState through the object property dicv:hasPropertyState. The use of property states makes it possible to associate properties with different values at different times and coming from different origin.

For the objectification of quantitative properties there are classes dicv:QuantitativeProperty and dicv:QuantitativeState that are subclasses of dicv:Property and dicv:PropertyState, respectively. They allow the connection of a quantity kind (dicv:QuantityKind) to a property as well as the unit of measurement (dicv:Unit).

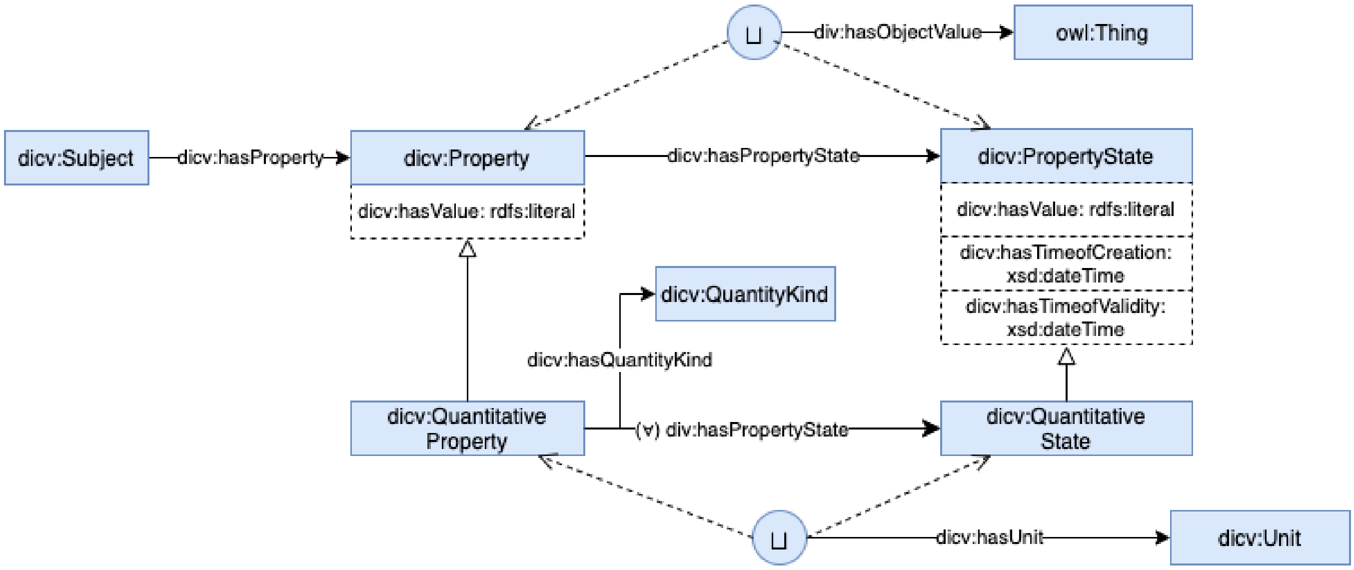


**Figure 14: Objectified properties in DiCon**

Figure 15 presents an example of the objectification of a property. There is an object :scenario1 of the type dici:Scenario that has the datatype property dices:hasInvestmentPaybackTime with the value "10"^^xsd:decimal. This can be represented as a simple RDF triple. When more information about this property needs to be expressed, an object of type dicv:QuantitativeProperty is created and the :scenario1 is connected to it through dicv:hasProperty relation. Now it is possible to associate to the property object a quantity kind object (of type dicv:QuantityKind) through the relation dicv:hasQuantityKind.

The property object can be associate a value directly, through the datatype property dicv:hasValue, and also the unit of measure, through the object property dicv:hasUnit. However, when the evolution of the values need to be captured, the property objects can be associated with property states, and in this case specifically with a dicv:QuantitativeState. A quantitative state allows the representation of a value (dicv:hasValue), unit (dicv:hasUnit) and time (dicv:hasTimeOfCreation).

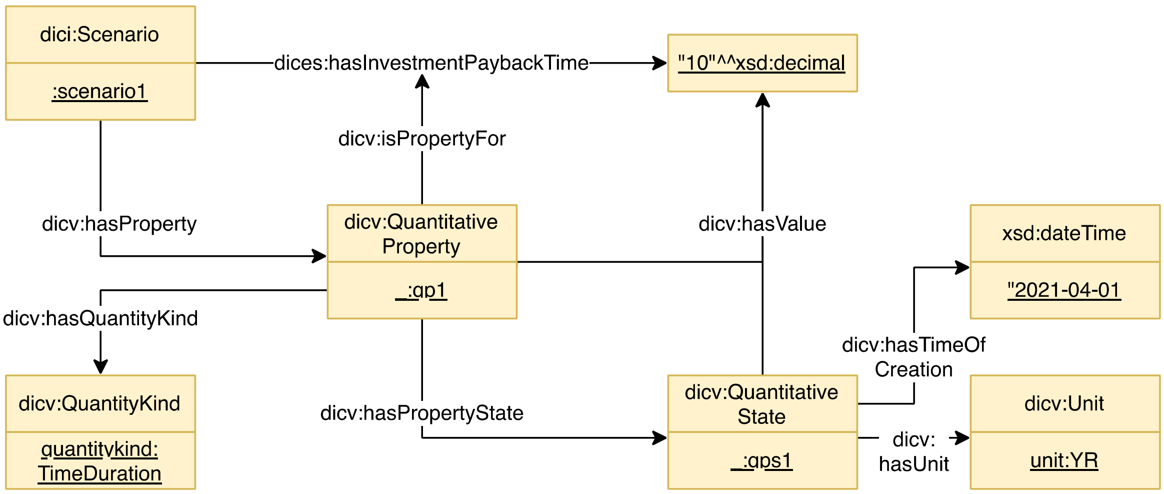


**Figure 15: Instance level examples of property objectification**

### 4.3.1 Classification pattern for properties

In DiCon ontologies the following pattern is used. Whenever there is a quantitative datatype property that needs to be objectified (such as dices:hasAnnualCoolingEnergyConsumption), a corresponding subclass of dicv:QuantitativeProperty is defined as follows:

```
:AnnualCoolingEnergyConsumption rdf:type owl:Class ;
   owl:equivalentClass [ rdf:type owl:Restriction ;
                    owl:onProperty dicv:isPropertyFor ;
                    owl:hasValue :hasAnnualCoolingEnergyConsumption ] ;
   rdfs:subClassOf :EnergyConsumptionQuality ,
               [ rdf:type owl:Restriction ;
                 owl:onProperty dicv:hasQuantityKind ;
                 owl:hasValue dicu:EnergyPerAreaTime ] ,
               [ rdf:type owl:Restriction ;
                 owl:onProperty dicv:hasUnit ;
                 owl:hasValue dicu:KiloW-HR-PER-M2-YR ] .
```

Whenever the datatype property :hasAnnualCoolingEnergyConsumption is objectified (e.g., using an instance from a class dicv:Property, the ontology reasoning is able to infer that the property object in fact belongs to the class :AnnualCoolingEnergyConsumption that has a specific quantity kind dicu:EnergyPerAreaTime and specific unit of measure dicu:KiloW-HR-PER-M2-YR.

The objectification concepts shown in the Figure 14 are defined in the Digital Construction Variables ontology. The concepts themselves are not new; the basic concepts have been presented earlier in the Ontology for Property Management (Rasmussen, 2018) almost with the same names . They also correspond to how entities, properties and values are modeled in established ontologies such as QUDT, SSN/SOSA and Saref Core.

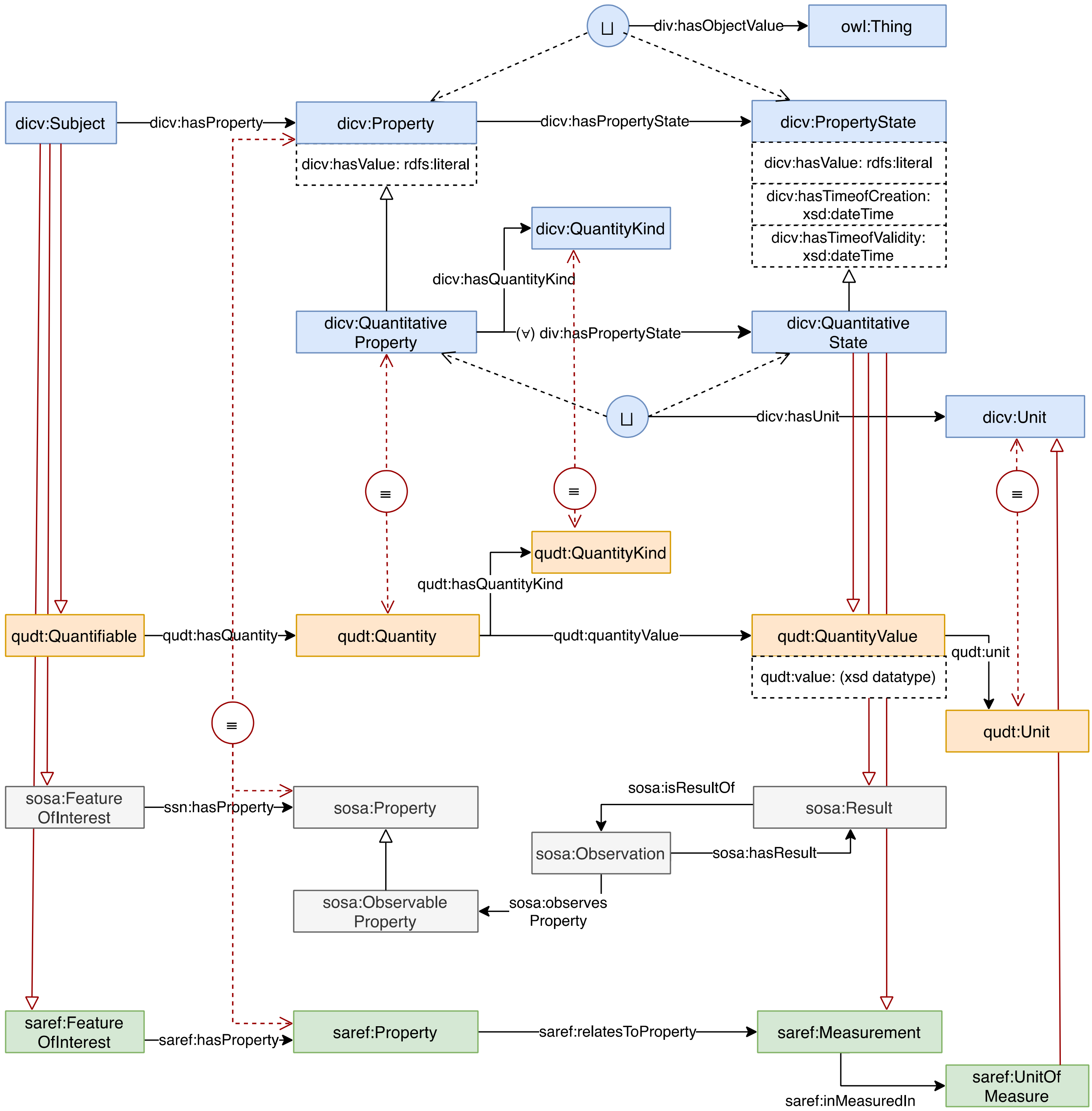


**Figure 16: Alignment of property objectification with QUDT, SSN/SOSA and Saref Core**

### 4.3.2 Alignment with QUDT, SSN/SOSA and Saref Core

The main parts of the alignment of the property objectification concepts of Digital Construction Variables with QUDT, SSN/SOSA and Saref Core is shown 16. The full alignment is available in the https://w3id.org/digitalconstruction/0.5. The basic guidelines are the following:

- Digital Construction Variables (dicv:) ontology has the root concept dicv:Subject which is only needed in the ontology itself for alignment purposes. It is aligned as a subclass of the similar root concepts in other ontologies: qudt:Quantifiable, sosa:FeatureOfInterest and saref:FeatureOfInterest.

- All ontologies have a concept for a property, corresponding dicv:Property. Concepts sosa:Property and saref:Property can be considered equivalent with dicv:Property and qudt:Quantity equivalent with dicv:QuantifiableProperty. In SOSA there is a concept called sosa:ObservableProperty which is difficult to align with any concept in the other ontologies: observability does not imply quantifiability nor the lack of it.
- All ontologies have a concept to represent a value at some point. The dicv:QuantityState can be considered as subclass of qudt:QuantityValue, sosa:Result and saref:Measurement. It can be seen as specialization those concepts.
- The concept dicv:Unit is equivalent with qudt:Unit. Since saref:UnitOfMeasure is based on more limited unit ontology than QUDT and since the concept “unit of measure” is more limited than the units in QUDT which can also contain, for instance, currency units, it has been considered correct to align saref:UnitOfMeasure to be a subclass of dicv:Unit.
- The object properties and datatype properties are aligend to comform to the alignment of the main concepts.

### 4.3.3 Relation to RDF reification

Finally, it should be noted that the objectified properties of DiCon can be aligned with the RDF reification terminology (Hayes, 2014) that consist of four concepts: rdf:Statement, rdf:subject, rdf:predicate, and rdf:object. Reification allows to create an object of type rdf:Statement to represent an RDF triple. The properties rdf:subject, rdf:predicate, and rdf:object (each defined as an instance of rdf:Property) indicate the different parts of a triple. The reason for the existence of the reification terminology in RDF is the need to connect additional information to a triple, related to time, provenance, certainty, etc. The correspondences of the objectified properties of Digital Construction Variables with the reification terminology is shown in Figure 17. These correspondences cannot be formally used as an alignment, since rdf:Property is not understood by OWL.

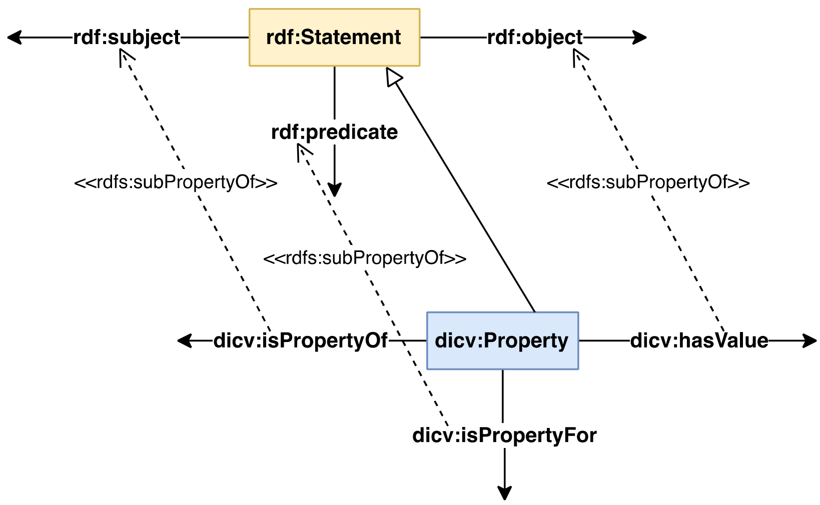


**Figure 17: Correspondences of objectified properties of DiCon with RDF reification terminology**

The significance of these correspondences deals with the RDF-star/SPARQL-star enhancement (Hartig, 2017) that has a special syntax for reification. It has made reification much more usable and lead to better implementation support for it. RDF-star/SPARQL-star makes RDF databases much closer to property graph databases and it is already supported by many RDF database vendors. This alignment could provide a bridge by which DiCon ontologies could be made compatible with RDF-star/SPARQL-star specification in the future.

### 4.3.4 Constraints on the properties

An important part of the knowledge about renovation management can be represented as constraints between properties of entities. In the model presented in the Digital Construction Variables ontology, any property can be associated with constraints, some of which can apply just to the particular property, and some are between two or more variables.

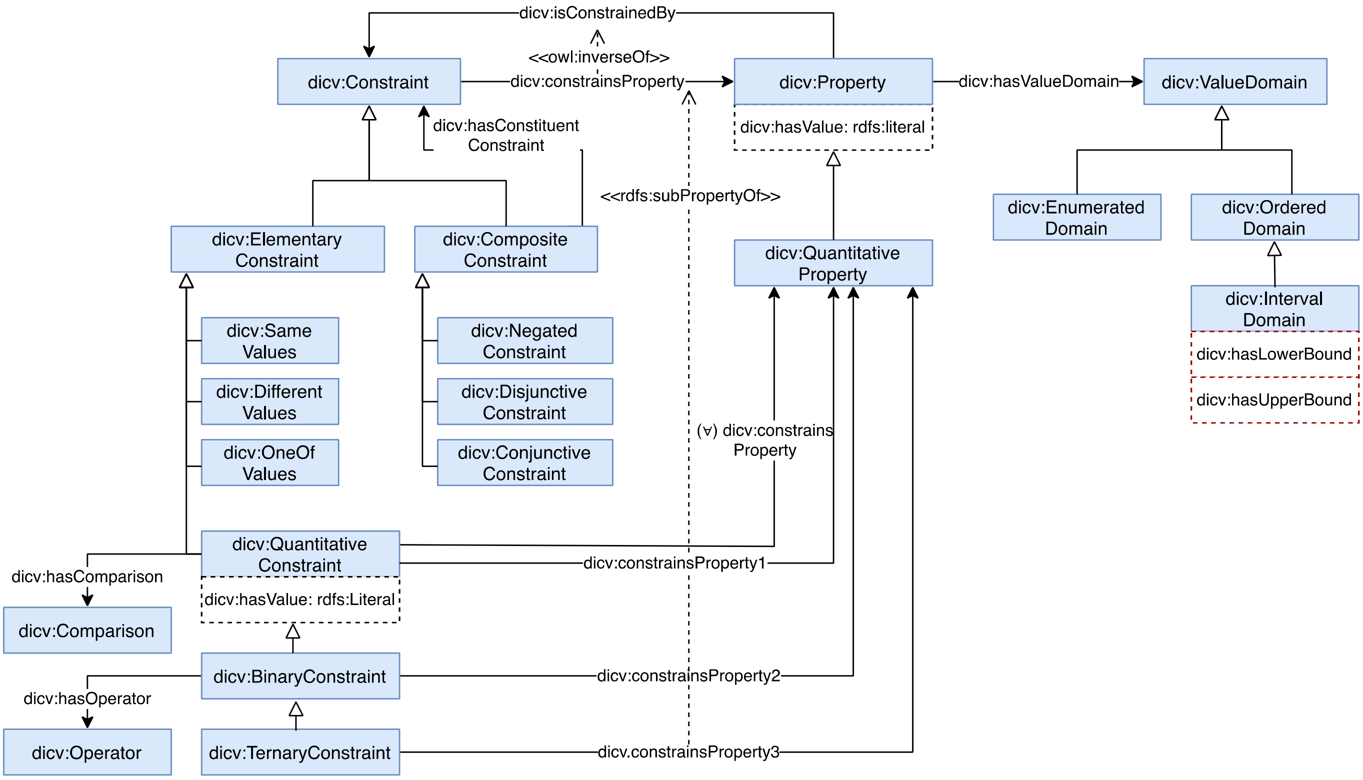


**Figure 18: Constraints associated with properties**

## 4.4 Information

ISO 19650 defines the practices for “information management using building information modelling”. From the perspective of an information management system, the main concepts are the Information Model and Information Container. Information Container corresponds roughly to a file containing a dataset of some kind. The Information Model is a set of Information Containers organized in a hierarchical fashion to a so-called federation or information container breakdown structure.

The main classes of Digital Construction Information ontology are shown in Figure 19. The class dici:InformationContentEntity represents all identifiable information contents. It has the relation (an object property) dici:isAbout to entities to capture the fundamental characteristic of information that it is always about something. Information content entities can be stored in multiple different physical media and locations, whose identities are not usually interesting. However, in case there is a need to refer to primary source of the information this capability is in the ontology. Information content entity can be related to a physical file (dici:File) with the object property dici:hasCarrier or to a data service (dici:DataService) with object property dici:isServedBy.

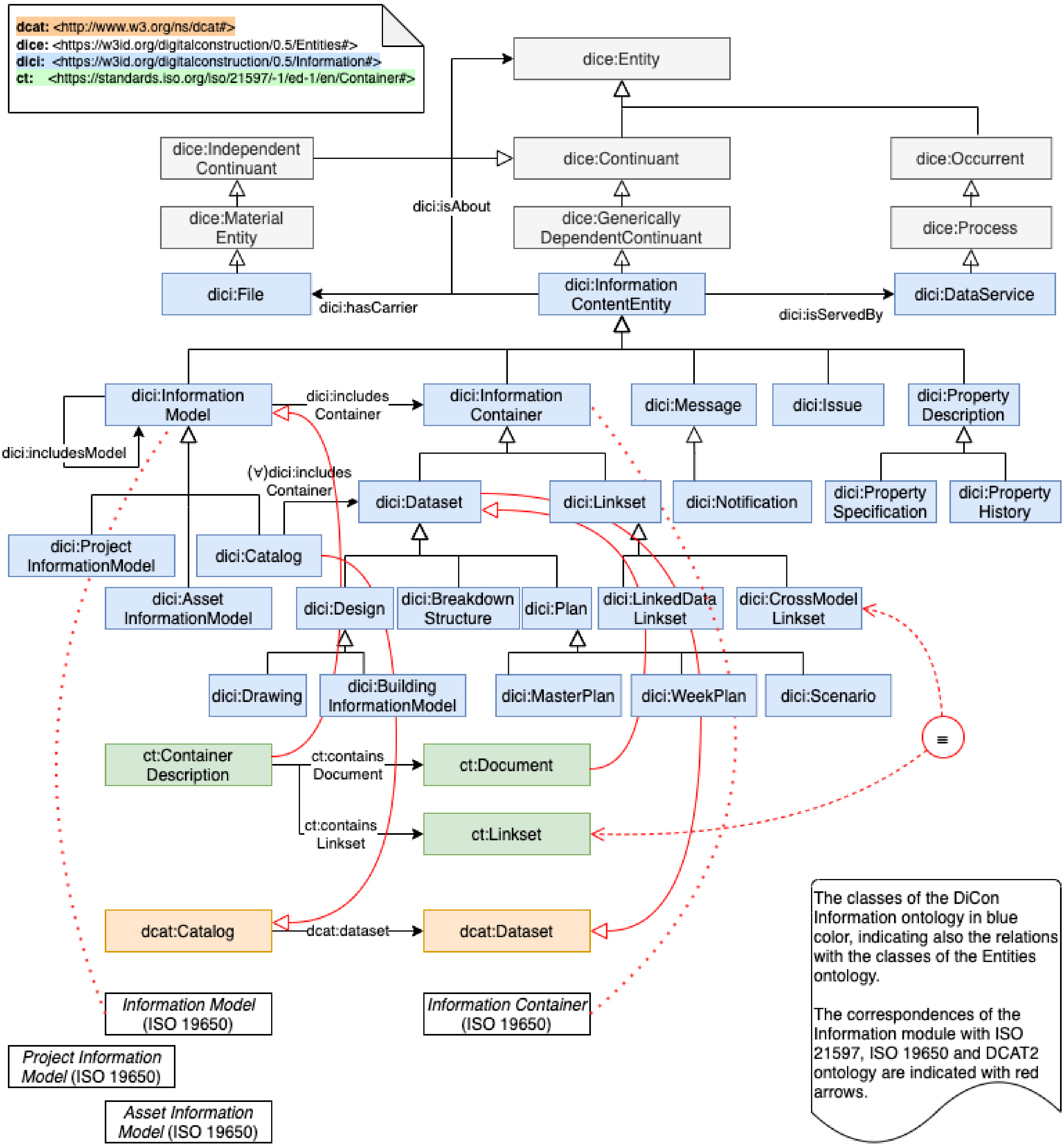


**Figure 19: Main classes of Digital Construction Information Ontology**

### 4.4.1 Information models, information containers and contexts

From the information management perspective, the most relevant information content entities are dici:InformationModel that corresponds to the concept *Information Model* and dici:InformationContainer that corresponds to the concept *Information Container* in the ISO 19650. Since ISO 19650 describes the terms of *Federation* and *Information Container Breakdown Structure*, the ontology specifies two relations:

dici:includesContainer that allows to specify the set of containers that a model includes. As an implementation approach for the hierarchical structure of the information model (as illustrated in Figure 3), any information model can include submodels – that is, instances of dici:InformationModel – through the relation dici:includesModel.

The main types of dici:InformationContainer are dici:Dataset and dici:Linkset. The interesting entities with respect to workflow management are the subclasses of dici:Design (dici:BuildingInformationModel, dici:Drawing), the subclasses of dici:Plan (dici:MasterPlan and dici:WeekPlan) and the class dici:BreakdownStructure.

### 4.4.2 Alignment with ICDD and DCAT2

The Figure 19 also shows in the red color the key correspondences of the Information module with ICDD and DCAT2; the full alignments are available in the Deliverable 3.6 (BIM4EEB-D36, 2022). The ISO 21597 ICDD container ontology (ct:) is aligned as follows. The ct:ContainerDescription is subclass of dici:InformationModel, the ct:Document is a subclass of the dici:Dataset and ct:Linkset is equivalent to dici:CrossFileLinkset. From the DCAT2 catalogue ontology the dcat:Catalog is superclass of dici:InformationModel and dcat:Dataset the superclass of dici:Dataset.

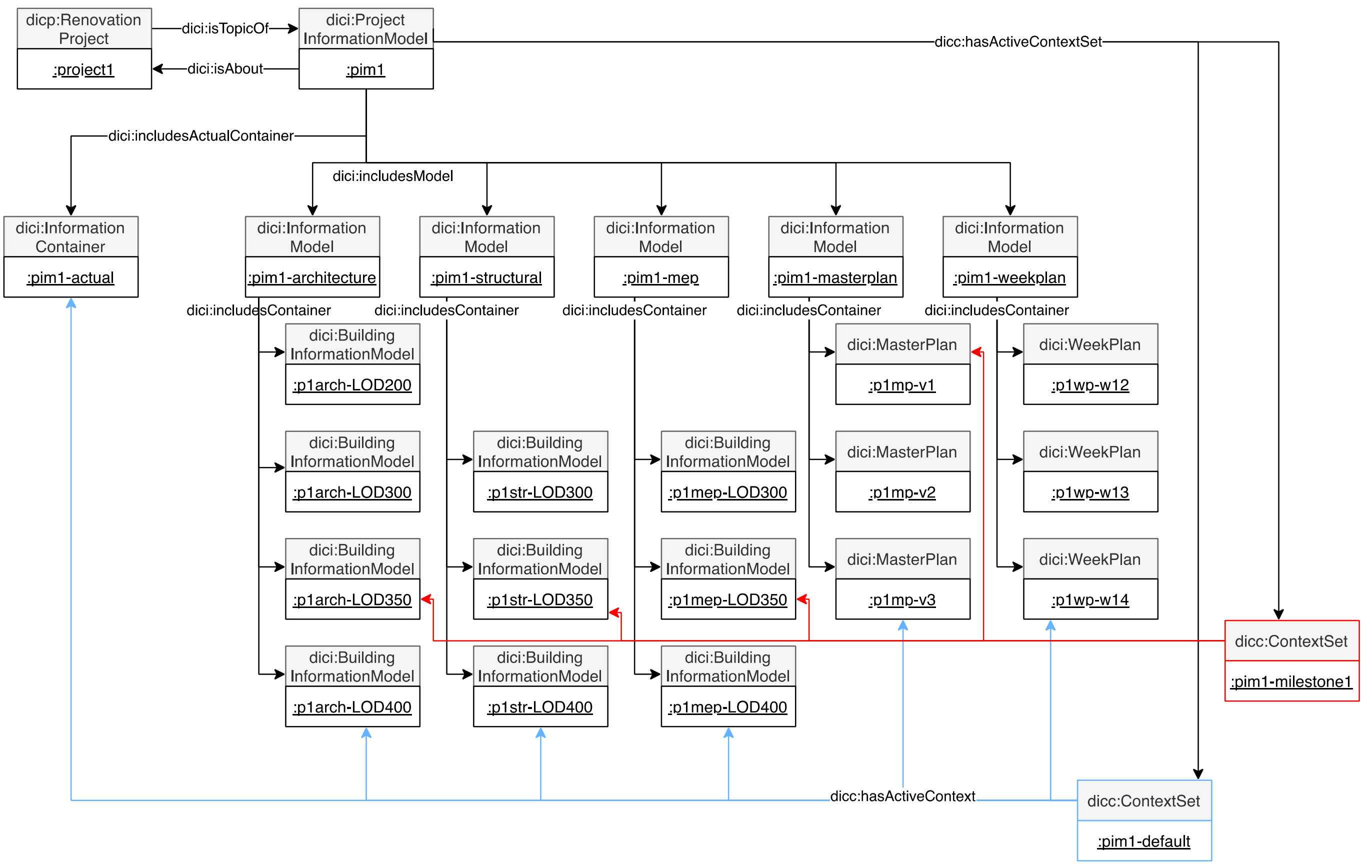


**Figure 20: An example of the structure of a project information model**

### 4.4.3 Example federation of an information model

Figure 20 shows an example of a project information model of a renovation project, based on the discipline-based federation approach, as described in (ISO19650-1, 2018). The figure show a project :project1 that is associated with a project information model :pim1. At the top level there is just one container that is related to the information model with object property dici:includeActualContainer that is a subproperty of dici:includesContainer. It should be noted that an information model can contain two types containers:

- *Actual container* contains things that have happened in the project or asset management. It is similar to log files in that it represents the *past*. Examples are events such as start of an activity, results of observations, publication of a new plan (meta information), and so on. Usually there is a need for just one actual container associated to a project, since there is only one past.
- *Modal containers* contain things that are planned, designed, or envisioned to take place in the *future*. As a rule these things never happen as thought. These containers include the contents of designs, plans, renovation scenarios, etc. There can be alternative scenarios, several versions of plans, and so on. Most of modal containers end up being obsolete at some point of a project.

The information model is organized hierarchically; it consists of sub-information models at the intermediate levels of the hierarchy and information containers at the leaf levels. The hierarchy is based on the disciplines: there are separate branches for architectural, structural, and MEP models as well as for master and week plans.

To be able to use the information in an information model in an efficient manner, there needs to be pointers to the active containers, that is, the containers that have not become obsolete yet. This is achieved by active context sets associated with the information model. In the Figure 20 there are two such context sets. One is a default context set that contains the most recent containers of different disciplines (shown in blue color). The other one is a context set that indicates the containers that were active at the milestone of a project (shown in red).

It should be noted that the dici:BuildingInformationModel containers whose name suggests a particular LOD level – for instance, :p1arch-LOD350 in Figure 20 – contain BIM models that have LOD information up to that level. Since different entities in a same model can be at a different LOD, in a LOD350 model there can also be some information at LOD300 or LOD200.

## 4.5 Agents

The Digital Construction Agents contain the definitions of the agents and their relations to activities. The relations of the concepts to the upper structure of the ontology in Entities is shown in Figure 12. In Figure 21 the main agent-related classes are shown. The basic class is dica:Agent, under which there are subclasses dica:Person and dica:Organization. Teams (dica:Team) are special type of information organizations and its subclasses dica:DeliveryTeam and dica:AssetOperatingTeam are terms derived from ISO 19650.

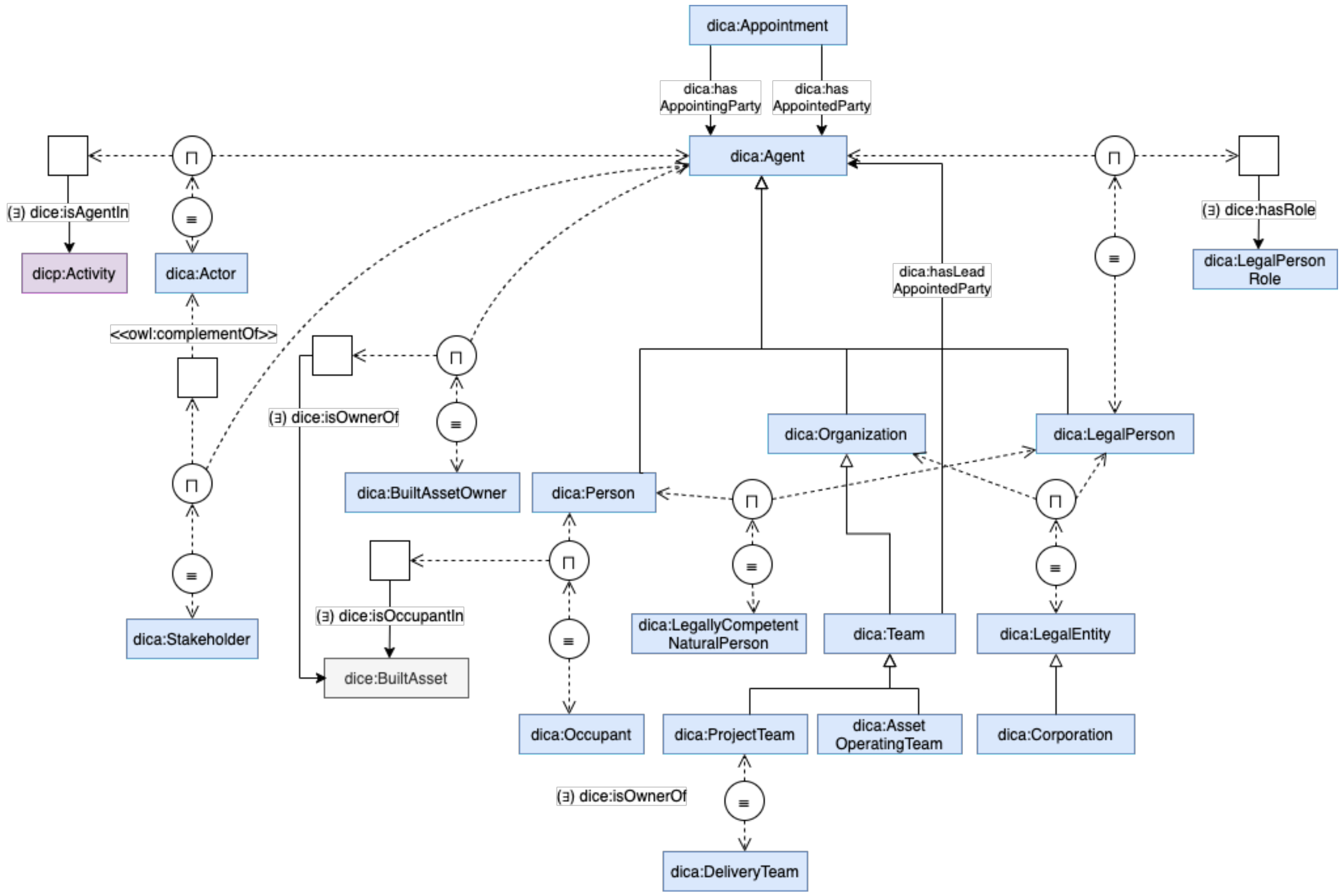


**Figure 21: Main classes of Digital Construction Agents**

### 4.5.1 Appointments and project team

Figure 22 shows an example of a structure of a project team where one party is the lead appointed party that has been appointed by a built asset owner, and who has further appointed other members to the team. The concepts have been defined as specified in the ISO 19650 terminology.

The model enables the representation of teams both with and without internal appointment structures. For instance, it is possible to represent the project team of a project using the Integrated Project Delivery approach by defining those parties that belong the core building team - owner, architect, general contractor, building engineers, and so on – as one team without an internal appointment structure. This core team can then make further appointments with external parties.

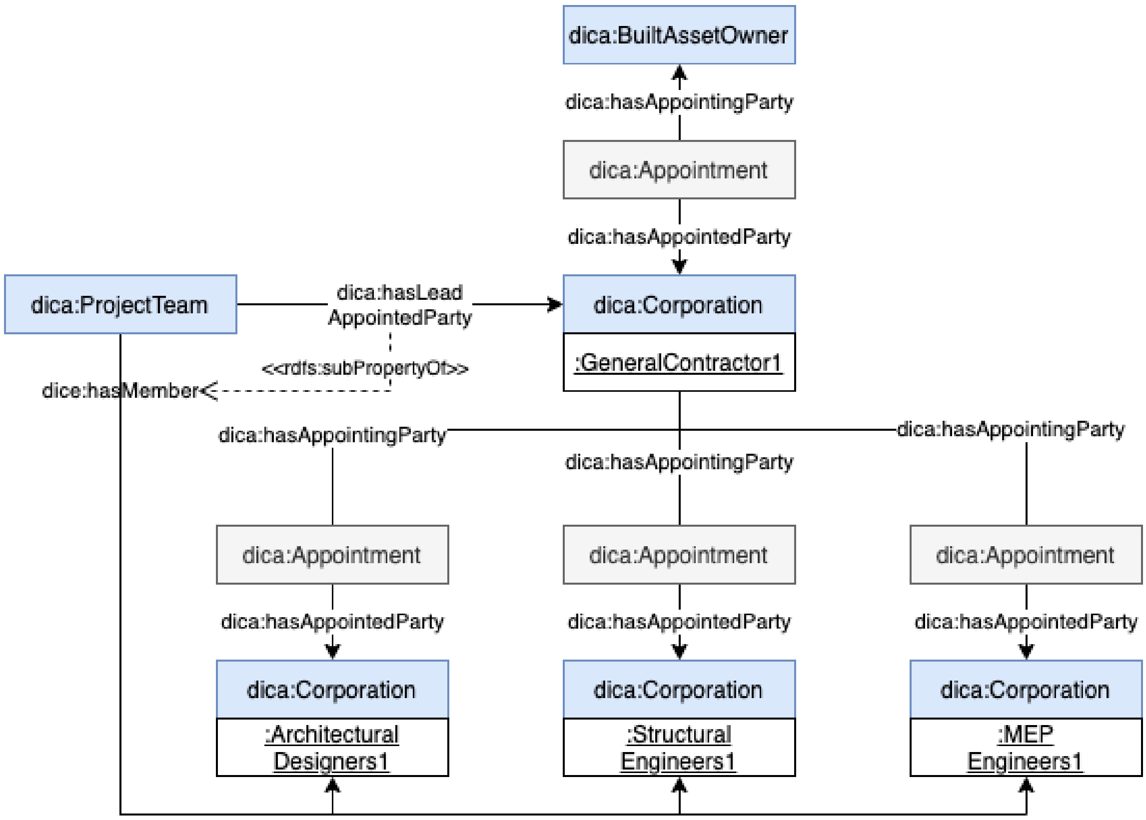


**Figure 22: Example of a project team**

### 4.5.2 Legal persons, owners, and occupants

To define other relevant concepts in a proper manner, a concept of dica:LegalPersonRole is defined as a subclass of dice:Role. To be a legal person is not an inherent character of any entity but rather a social or juridical role. Therefore, the role is the central concept to enable other legally oriented definitions.

Legal person dica:LegalPerson is defined as an intersection of dica:Agent and someone having dica:LegalPersonRole as some values of the property dice:hasRole. The following definitional classes have been created:

- dica:LegalPerson = dica:Agent and
  (dice:hasRole some dica:LegalPersonRole)
- dica:LegalEntity = dica:Organization and dica:LegalPerson
- dica:LegallyCompetentNatualPerson = dica:Person and dica:LegalPerson
- dica:BuiltAssetOwner = dica:LegalPerson and
  (dice:isOwnerOf some dice:BuiltAsset)
- dica:Occupant = dica:Person and
  (dica:isOccupantIn some dice:BuiltAsset)
- dica:Actor = dica:Agent and (dica:isAgentIn some dicp:Activity)
- dica:Stakeholder = dica:Agent and (not dica:Actor)

A corporation (dica:Corporation) is a subclass of dica:LegalEntity. There are also other kinds of legal entities, not represented here. The legal concepts have been defined following the FIBO ontology (Bennet, 2013).

## 4.6 Processes

The activities are connected to different classes around the overall ontology in the manner shown in Figure 23. The ontology defines the concept of dicp:ResourceRole which is a primary concept to determine what is a resource. To be a resource is not the inherent characteristic of an entity but a role that different entities can play at different times with respect to activities. An entity can have a resource role with respect to multiple different activities. If an entity is in a resource role with respect to some activity, it can be classified as a resource. Resource is therefore defined as follows:

dicp:Resource = dice:Entity and (dice:hasRole some dicp:ResourceRole)

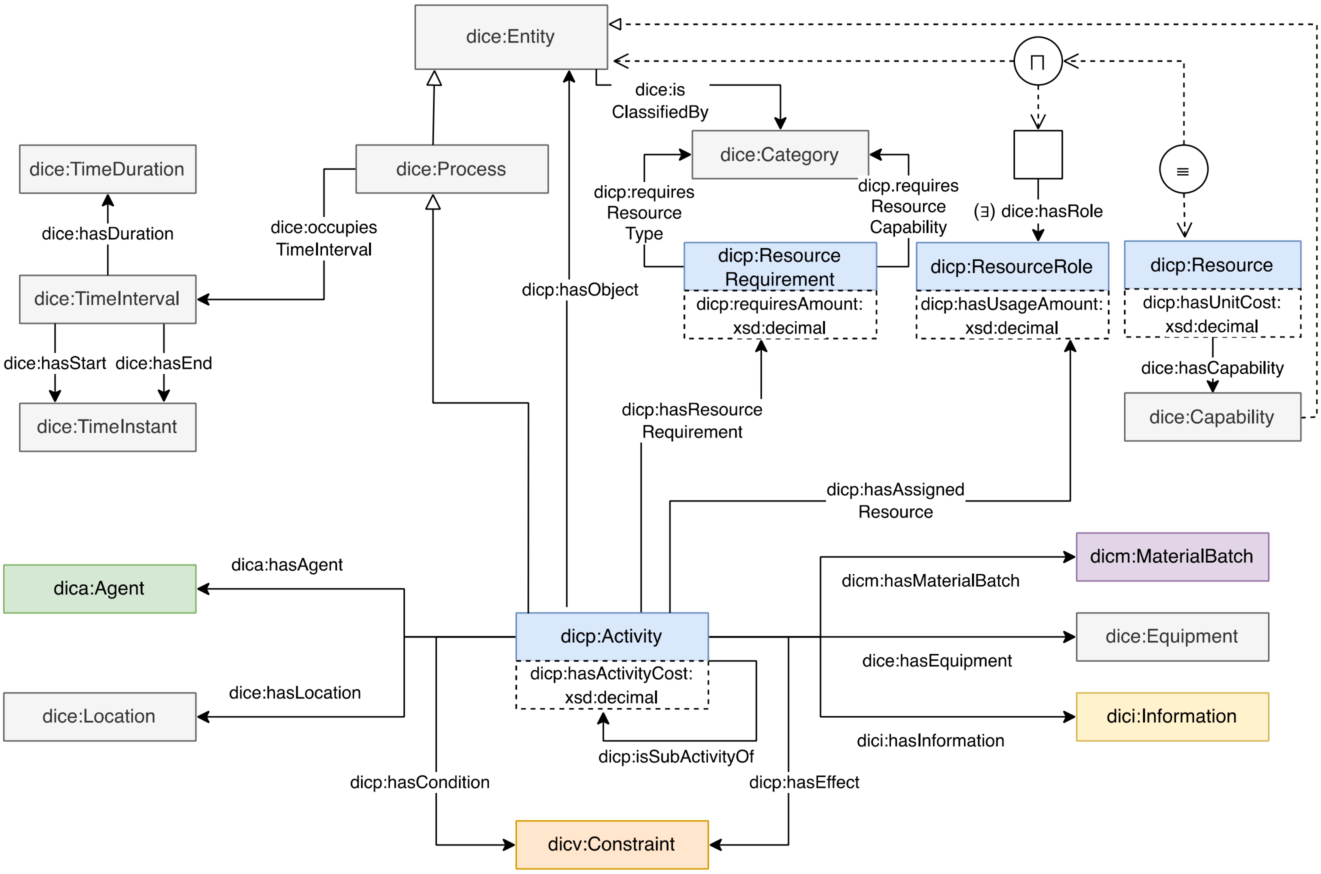


**Figure 23: Activities, flows and resources**

Resource role is associated with an activity through the relation (dicp:isResourceIn, inverse: dicp:hasAssignedResource). The resource role is thus always with respect to an activity. The resource role can further be specified with the datatype property dicp:hasUsageAmount, indicating the amount the activity uses the resource. Resource has a unit cost, which enables to compute the cost of the resource use in the activity.

The activities can have resource requirements (dicp:ResourceRequirement) that specify the category (dice:Category) of the resource needed or a category of the capability (dicp:Capability) of a resource needed.

# 5 Ontology use

This section describes the use of the ontology: how definitions can be accessed, and how data is organized according to the ontology.

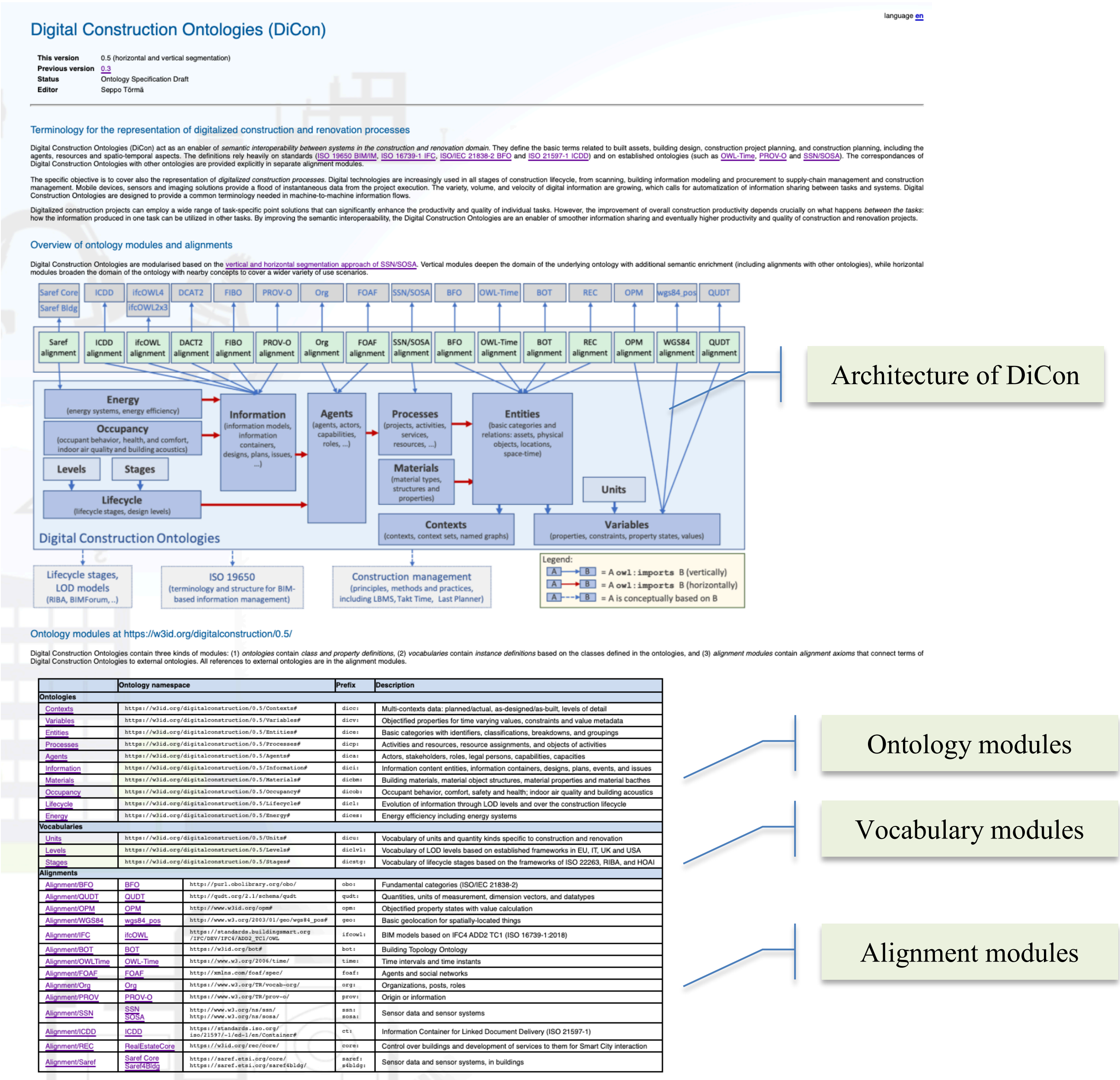


**Figure 24: Landing page of DiCon**

## 5.1 Accessing the ontology

All modules of Digital Construction Ontologies have been defined in OWL, the documentation has been created with pyLODE. Both the documentation and the ontology definition files have been published in the following address (the page is shown in Figure 24):

https://w3id.org/digitalconstruction/0.5

The ontology modules described in this deliverable are the following:

**Table 8: The addresses of the ontology modules**

| | |
|---|---|
| **Variables** | https://w3id.org/digitalconstruction/0.5/Variables |
| **Contexts** | https://w3id.org/digitalconstruction/0.5/Contexts |
| **Entities** | https://w3id.org/digitalconstruction/0.5/Entities |
| **Processes** | https://w3id.org/digitalconstruction/0.5/Processes |
| **Agents** | https://w3id.org/digitalconstruction/0.5/Agents |
| **Information** | https://w3id.org/digitalconstruction/0.5/Information |

The ontologies are maintained in separate repositories in GitHub at the following address:

https://github.com/digitalconstruction

Each ontology module is in a separate GitHub repository in that address. For instance, Entities modules can be found from the address:

https://github.com/digitalconstruction/Entities

The repository also contains all the alignments described in this report in subdirectories of the following repository:

https://github.com/digitalconstruction/Alignments

## 5.2 Data organization

A small example of the organization of the data according to the Digital Construction Ontologies to different graphs in an RDF Dataset is shown in Figure 25. The example is a limited and simplified one, since it only shows 22 entities and related statements in 7 different graphs, while in a complete renovation project there can be tens of thousands of entities located in hundreds of different graphs.

The example is drawn according to the Chowlk specification but since Chowlk only supports OWL constructs and does not specify any visual representation for RDF graphs, the graphs are indicated with white rectangles. The rectangles show which statements (quads) are located in which graph. It should be noted that the entities do not belong to any particular graph; only the statements belong to graphs. Entities are just URI identifiers, and the statements of every graph can refer to any URI.

Data is assumed to be maintained in an RDF database in one repository that hosts an RDF dataset. Data is organized to the graphs of the RDF dataset in the following manner:

- Default graph contains a catalog of entities (dici:Catalog) that are represented in this particular RDF dataset. These entities are independent, non-information entities: projects, assets and agents. The catalog consist of header objects of type dici:Dataset (or one of its subclass) each of which describes the entity, and contains a reference to the entity and to the named graph in which the data about the entity resides. It may also contain additional information about the access rights, time period, geographical location, and so on that help the user find the right project and determine whether the user is authorized to access it. In the example, there is an object :DefaultCatalog that includes four different datasets, one for a project, one for a built asset, and two for agents.
- The content of each dataset is represented in its own named graph. This graph contains the current and historical information about the entity. It can be thought of as a log of things that have happened regarding the entity. In the example some minimal contents of these graphs are shown.

In addition to the data about an entity, the graph can represent additional information entities, such as :PIM1 of the type dici:ProjectInformationModel in the example. If these information entities belong to some subclass of dici:InformationContainer, the content of each of these is represented in its own named graph, as in example the :R1BIM1 of type dici:BuildingInformationModel or the :R1MP1 of the type dici:Masterplan.

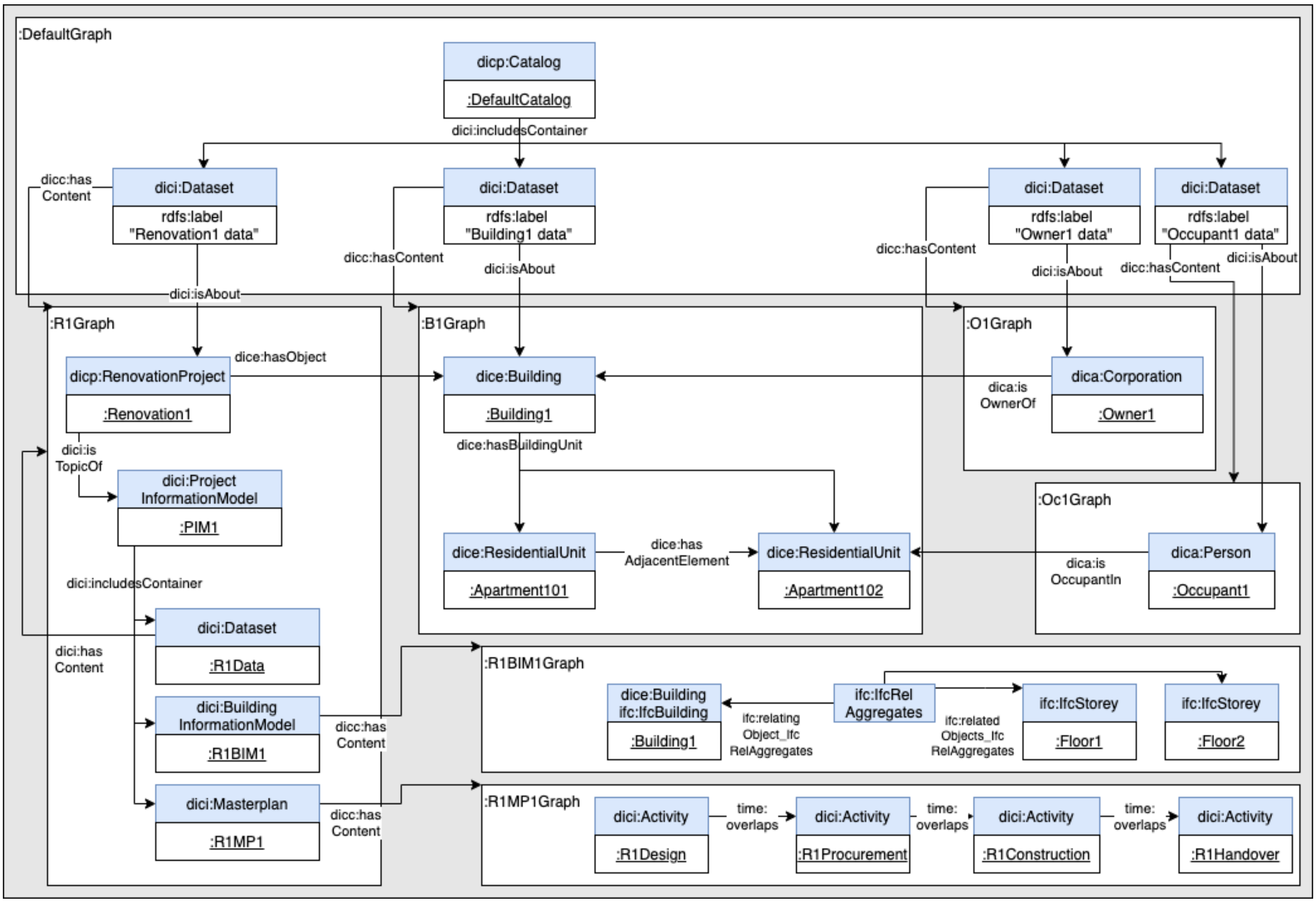


**Figure 25: Example of data organization in an RDF Dataset**

The example in Figure 25 shows a small sample of entities at a particular point of project execution. During the execution the amount of data grows and becomes much more complex.

**Table 9: TriG serialization of the example**

```
@prefix :     <http://example.com/id/> .
@prefix xml:  <http://www.w3.org/XML/1998/namespace> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ifc:  <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL> .
@prefix dica: <https://w3id.org/digitalconstruction/0.5/Agents#> .
@prefix dicc: <https://w3id.org/digitalconstruction/0.5/Contexts#> .
@prefix dice: <https://w3id.org/digitalconstruction/0.5/Entities#> .
@prefix dicp: <https://w3id.org/digitalconstruction/0.5/Processes#> .
@prefix dicv: <https://w3id.org/digitalconstruction/0.5/Variables#> .
@prefix dici: <https://w3id.org/digitalconstruction/0.5/Information#> .
```

```
:DefaultCatalog a dici:Catalog ;
   dici:includesContainer _:Dataset1,
      _:Dataset2, _:Dataset3, _:Dataset4 :

_:Dataset1 a dici:Dataset ;
   rdfs:label "Renovation1 data" ;
   dici:isAbout :Renovation1 ;
   dicc:hasContent :R1Graph .

_:Dataset2 a dici:Dataset ;
   rdfs:label "Building1 data" ;
   dici:isAbout :Building1 ;
   dicc:hasContent :B1Graph .

_:Dataset3 a dici:Dataset ;
   rdfs:label "Owner1 data" ;
   dici:isAbout :Owner1 ;
   dicc:hasContent :O1Graph .

_:Dataset4 a dici:Dataset ;
   rdfs:label "Occupant1 data" ;
   dici:isAbout :Occupant1 ;
   dicc:hasContent :Oc1Graph .


GRAPH :R1Graph {
   :Renovation1 a dicp:RenovationProject ;
      dice:hasObject :Building1 ;
           dici:isTopicOf :PIM1 .

   :PIM1 a dici:ProjectInformationModel ;
      dici:includesContainer :R1Data ,
        :R1BIM1, :R1MP1 .

   :R1Data a dici:Dataset ;
      dici:hasContent :R1Graph .

   :R1BIM1 a dici:BuildingInformationModel ;
      dici:hasContent :R1BIM1Graph .

   :R1MP1 a dici:Masterplan ;
      dici:hasContent :R1MP1Graph .
}
```

```
GRAPH :B1Graph {
   :Building1 a dice:Building ;
      dice:hasBuildingUnit :Apartment101,
        :Apartment102 .

   :Apartment101 a dice:ResidentialUnit .
   :Apartment102 a dice:ResidentialUnit ;
      dice:hasAdjacentElement :Apartment1 .
}

GRAPH :O1Graph {
   :Owner1 a dica:Corporation ;
      dica:isOwnerOf :Building1 .
}

GRAPH :Oc1Graph {
   :Occupant1 a dica:Person ;
      dica:IsOccupantIn :Apartment102 .
}

GRAPH :R1BIM1Graph {
   :Building1 a ifc:IfcBuilding .
   :B1Floor1 ifc:IfcBuildingStorey .
   :B1Floor2 ifc:IfcBuildingStorey .
   _:aggregate a ifc:IfcRelAggregates ;
      ifc:relatingObject_IfcRelAggregates
        :Building1 ;
          ifc:relatedObjects_IfcRelAggregates
        :B1Floor1, :B1Floor2 .
}

GRAPH :R1MP1Graph {
   :R1Design a dicp:Activity ;
      time:overlaps :R1Procurement .

   :R1Procurement a dicp:Activity ;
      time:overlaps :R1Construction .

   :R1Construction a dicp:Activity ;
      time:before :R1Handover .

   :R1Handover a dicp:Activity .
}
```

## 5.3 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 present the alignments with external ontologies, and address the competency questions, their formalization in SPARQL according to the terms of DiCon and evaluation with test data.

# 6 Summary

This deliverable described a set of interlinked ontology modules belonging to Digital Construction Ontologies (DiCon): Contexts, Variables, Entities, Processes, Agents and Information.

The goal of this effort is to support the representation of renovation processes for the purpose of construction management in renovation and construction projects.

The ontologies take into account the various dependencies and constraints in construction phase by covering the entities faced in construction management - building objects, information objects, delivery teams, task teams, locations and workspaces, equipment, materials, and external conditions – and their connections to activities.

The ontology also addresses the dynamic evolution of information during project execution: designs go through several levels of development, planning is carried out at multiple levels, and plans are also periodically adjusted. The information can be represented at multiple different contexts and additional metadata can be provided about the information, enabling advanced BIM change management.

The ontology makes definition according to the terminology of ISO 19650 to support a Linked Data approach for ISO 19650 compatible information management. The BIM change management is supported though the ISO 19650 concepts of information model and information container. Interesting future directions for this work would b, firstly, to provide representational support for remote containers within an information model, and secondly, to develop mediators that can covert non-RDF data on-the-fly to RDF.

The ontology is aligned with IFC (ISO 16739) and BFO (ISO/IEC 21838) as well as with many established ontologies, such as SSN/SOSA and Saref Core/Buildings for sensor data, OWL-Time for temporal entities, QUDT for units of measurement, FOAF and Org for agents and organizations, and PROV-O for provenance information.

An interesting future possibility is to study the impact that RDF*/SPARQL* could have on the representation of reified or objectified properties in DiCon. This would be especially relevant, if RD*/SPARQL* will have a large impact on the implementation techniques of RDF databases, and the performance of reification.

# 7 Bibliography

[Allen, 1983] Allen, J.F., 1983. Maintaining knowledge about temporal intervals. Communications of the ACM, 26(11):832-843.

[Almeida, 2012] Almeida, M. B., Slaughter, L., & Brochhausen, M. (2012). Towards an ontology of document acts: Introducing a document act template for healthcare. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 420-425). Springer, Berlin, Heidelberg

[Akinci, 2002] Akinci, B., Fischer, M., & Kunz, J. (2002). Automated generation of work spaces required by construction activities. *Journal of construction engineering and management*, 128(4), 306-315.

[Abels, 2006] Abels, S., Ahlemann, F., Hahn, A., Hausmann, K., & Strickmann, J., 2006. PROMONT – A project management ontology as a reference for virtual project organizations. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 813-823). Springer, Berlin, Heidelberg.

[Arp, 2015] Arp, R., Smith, B., & Spear, A. D. (2015). *Building Ontologies with Basic Formal Ontology*. Mit Press.

[Ballard, 2000] Ballard, G., 2000. The last planner system of production control, Doctoral dissertation, University of Birmingham.

[Barbosa, 2017] Barbosa, F., Woetzel, J., Mischke, J., Ribeirinho, M., Sridhar, M., Parsons, M., Bertram, N., et al., 2017. Reinventing construction: A route to higher productivity, McKinsey Global Institute, available at: http://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/reinventing-construction-through-a-productivity-revolution.

[Beetz, 2009b] Beetz, J., 2009. Facilitating distributed collaboration in the AEC/FM sector using Semantic Web technologies, Ph.D. dissertation, Eindhoven University of Technology, ISBN 978-90-6814-618-9.

[Bennet, 2013] Bennett, M. (2013). The financial industry business ontology: Best practice for big data. Journal of Banking Regulation, 14(3), 255-268.

[Berners-Lee, 2006] Berners-Lee, T., 2006. Linked Data – Design Issues, W3C Note, URL: http://www.w3.org/DesignIssues/LinkedData.html.

[BFO-ISO, 2019] BFO-ISO (2019), Basic Formal Ontology, OWL definition draft for ISO standard ISO/IEC:21838-2, fetched 2019-12-19. URL: https://buffalo.app.box.com/v/bfo-iso-owl-cl/folder/81126451877.

[BIM4EEB-D21, 2019] BIM4EEB (2019). Definition of relevant activities and involved stakeholders in actual and efficient renovation processes, BIM4EEB Deliverable 2.1, EU LC-EEB-02-2018, Project grant no 820660.

[BIM4EEB-D31, 2019] BIM4EEB (2019). A BIM-based framework for building renovation using the linked data approach and ontologies, BIM4EEB Deliverable 3.1, EU LC-EEB-02-2018, Project grant no 820660.

[BIM4EEB-D36, 2022] BIM4EEB (2022), Integrated Linked Data Modeling and Sharing Framework, Deliverable 3.6, Grant Agreement 820660 BIM4EEB. European Commission.

[BIMForum, 2018] BIMForum, 2018. Level Of Development (LOD), Specification Part I & Commentary. September 2018.

[Bizer, 2014] Bizer, C., & Cyganiak, R. (2014). RDF 1.1 TriG. W3C recommendation, 25 February 2014. URL: https://www.w3.org/TR/trig/.

[Brickley, 2014] Brickley, D., Guha, R. V., & McBride, B. (2014). RDF Schema 1.1. *W3C recommendation*, *25*, 2004-2014. URL: https://www.w3.org/TR/rdf-schema/

[Brickley, 2014b] Brickley, D., & Miller, L. (2007). FOAF vocabulary specification 0.99. W3C Namespace Document, W3C.

[Ceusters, 2015] Ceusters, W., & Smith, B. (2015). Aboutness: Towards foundations for the information artifact ontology. URL: https://philpapers.org/archive/CEUATF.pdf

[Cox, 2017] Cox, S. and Little, C., 2017. Time Ontology in OWL. W3C Recommendation. URL: https://www.w3.org/TR/owl-time/.

[Fox, 2002] Fox, M., & Long, D. (2002). PDDL+: Modeling continuous time dependent effects. In Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space (Vol. 4, p. 34).

[Fox, 2003] Fox, M., & Long, D. (2003). PDDL2. 1: An extension to PDDL for expressing temporal planning domains. Journal of artificial intelligence research, 20, 61-124.

[Garcia-Lopez, 2017] Garcia-Lopez, N., 2017. An Activity and Flow-based Construction Model for Managing On-site Work, Doctoral dissertation, Stanford University.

[Ghallab, 2016] Ghallab, M., Nau, D., & Traverso, P. (2016). *Automated planning and acting*. Cambridge University Press.

[Groth, 2010] Groth, P., Gibson, A., & Velterop, J. (2010). The anatomy of a nanopublication. *Information Services & Use*, *30*(1-2), 51-56.

[Grüninger, 1995] Grüninger, M., & Fox, M. S. (1995). The role of competency questions in enterprise engineering. In Benchmarking—Theory and practice (pp. 22-31). Springer, Boston.

[Haller, 2017] Haller, A., Janowicz, K., Cox, S., Le Phuoc, D., Taylor, K., Lefrançois, M. & Stadler, C. (2017). Semantic Sensor Network Ontology, W3C Recommendation 19 October 2017. URL https://www.w3.org/TR/vocab-ssn.

[Hartig, 2017] Hartig, O. (2017). RDF* and SPARQL*: An Alternative Approach to Annotate Statements in RDF. In International Semantic Web Conference, Posters, Demos & Industry Tracks.

[Hayes, 2014] Hayes, P.J and Patel-Schneider, P.F. (2014). *RDF 1.1 Semantics*, W3C Recommendation, W3C. URL: https://www.w3.org/TR/rdf11-mt/

[Heinonen, 2016] Heinonen, A., and Seppänen, O. (2016). “Takt Time Planning: Lessons for Construction Industry from a Cruise Ship Cabin Refurbishment Case Study” In: Proc. 24th Ann. Conf. of the Int’l. Group for Lean Construction, Boston, MA, USA, sect.2 pp. 23–32.

[Hollingsworth, 1995] Hollingsworth, D., and Hampshire, U. K. (1995). Workflow management coalition: The workflow reference model. Document Number TC00-1003, 19, 16.

[IOF, 2019] Industrial Ontologies Foundry (2019). URL: https://www.industrialontologies.org/, retrieved 2019-11-22.

[Kelley, 1959] Kelley Jr, J. E., & Walker, M. R. (1959, December). Critical-path planning and scheduling. Eastern Joint IRE-AIEE-ACM Computer Conference (pp. 160-173). ACM.

[Kenley, 2010] Kenley, R. and Seppänen, O., 2010. Location-based Management System for Construction: Planning, Scheduling and Control. Spon Press. London, New York.

[Koskela, 1999] Koskela, L. J., 1999. Management of production in construction: a theoretical view. 7th Annual Conference of the International Group for Lean Construction Berkeley, CA. 241–252.

[Koskela, 2000] Koskela, L., 2000. An exploration towards a production theory and its application to construction. VTT Publications 408, Espoo, Finland.

[Krafcik, 1988] Krafcik, J. F. (1988). Triumph of the lean production system. *MIT Sloan Management Review*, 30(1), 41.

[Lebo, 2013] Lebo, T., Sahoo, S. and McGuinness, D., 2013. PROV-O: The PROV Ontology, W3C Recommendation. URL: https://www.w3.org/TR/prov-o/.

[Nguyen, 2014] Nguyen, V., Bodenreider, O., & Sheth, A. (2014). Don't like RDF reification?: Making statements about statements using singleton property. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 759-770). ACM.

[Oraskari, 2016] Oraskari, J. and Törmä, S., 2016. Access Control for Web of Building Data: Challenges and Directions, in (eds.) Symeon Christodoulou, Raimar Scherer: eWork and eBusiness in Architecture, Engineering and Construction, CRC Press/Balkema.

[Poveda-Villalón, 2018] Poveda-Villalón, M., & Garcıa-Castro, R. (2018). Extending the SAREF ontology for building devices and topology. In Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018), Vol. CEUR-WS (Vol. 2159, pp. 16-23).

[Prud'hommeaux, 2014] Prud'hommeaux, E. and Carothers, G., 2014. RDF 1.1 Turtle – Terse RDF Triple Language. W3C Recommendation. URL: https://www.w3.org/TR/turtle/.

[Rasmussen, 2018] Rasmussen, M., Lefrançois, M., Bonduel, M., Hviid, C., & Karlshøj, J. (2018). OPM: An ontology for describing properties that evolve over time. In proc. of 6th *Linked Data in Architecture and Construction Workshop*. London, UK.

[Smith, 1996] Smith S., Lassila O., and Becker M. , 1996. A Scheduling Ontology: Informal Concept Descriptions, ICLL Working Paper, Carnegie Mellon University.

[Seppänen, 2010] Seppänen, O., Glenn, B. and Pesonen, S., 2010. The combination of Last Planner System and Location-Based Management System, Lean Construction Journal.

[Seppänen, 2014a] Seppänen O., 2014. A comparison of Takt time and LBMS planning methods. Proc. 22nd Annual Conf. of the Int'l. Group for Lean Constr. (IGLC 22), Oslo, Norway.

[Seppänen, 2014b] Seppänen, O., Evinger, J. and Mouflard M., 2014. Effects of the Location-Based Management System on Production Rates and Productivity. Construction Management and Economics, 32(6), pp. 608-624.

[Seppänen, 2019] Seppänen, Olli (2019). Personal communication.

[Tate, 1996a] Tate, A. (1996). Towards a plan ontology. AI* IA Notiziqe (Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on Aspects of Planning Research, 9(1), 19-26.

[Tate, 1996b] Tate, A. (1996). Representing plans as a set of constraints-the IN-OVA model. In Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96) (pp. 221-228). Edinburgh, Scotland, Morgan Kaufmann.

[Tate, 1998] Tate, A. (1998). Roots of SPAR—shared planning and activity representation. The Knowledge Engineering Review, 13(1), 121-128.

[Tate, 2001] Tate, A. (2001). IX and< INCA>: an Architecture and Related Ontology for Mixed-Initiative Synthesis Tasks. In In Proceedings of the Workshop on Planning/Scheduling and Configuration/Design.

[Törmä , 2013] Törmä, S., 2013. Semantic Linking of Building Information Models. IEEE 7th International Conference on Semantic Computing. IEEE, pp. 412–419.

[Törmä, 2014] Törmä, S., 2014. Web of Building Data Integrating IFC with the Web of Data. eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014; Mahdavi, A.; Martens, B.; Scherer, R., Eds. CRC Press, pp. 141–148.

[Vajda, 2019] Vajda, J., Merrell, E., & Smith, B. (2019). Toward an Ontology of Commercial Exchange. URL: https://philarchive.org/archive/VAJTAO

[Van der Aalst, 1998] Van der Aalst, W. M. (1998). The application of Petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01), 21-66.

[Wetherill, 2003] Wetherill, M., 2003. Knowledge management for the construction industry: the e-cognos project. *Journal of Information Technology in Construction (ITCon)*, *7*(12), 183-196.

[Womack, 1990] Womack, J. P., Jones, D. T., & Roos, D. (1990). *The machine that changed the world*, Rawson Associates. New York, 323.

[W3C OWL WG, 2012] W3C OWL WG, 2012. OWL 2 Web Ontology Language Document Overview (2nd ed). W3C Recommendation. URL: https://www.w3.org/TR/owl-overview/.

[W3C-SPARQL-WG, 2013] W3C SPARQL WG, 2013. SPARQL 1.1 Overview. W3C Recommendation. URL: https://www.w3.org/TR/sparql11-overview/.