Optimal Construction Management & Production Control

# D2.1 – Digital Building Twin Requirements Analysis and Data Model

WP2 – Digital Building Twin Process

Version 1.0

**Issue date:** 25/10/2021

**Author(s):** Nicolas BUS (CSTB), Sven OESAU (CSTB), Florent Coudret (CSTB), Jonas Schlenger (TUM), Timson Yeung (Technion), Mathieu THOREL (CSTB)

**Editor**: Nicolas PASTORELLY (CSTB)

**Lead Beneficiary**: CSTB

**Dissemination level**: Public

**Type** Report

**Reviewers:** André BORMANN (TUM), Rafael SACKS (TECHNION), Nicolas PASTORELLY (CSTB), Bruno FIES (CSTB), Markus König (RUB), Alan Redmond (CSTB)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 958398

## EXPLANATIONS FOR FRONT PAGE

**Author(s):** Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

**Editor:** Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author – if he has generated the Foreground – as well as an Editor – if he also edits the report on its own Foreground.

**Lead Beneficiary of Deliverable:** Only one. Identifies name of the partner that is responsible for the Deliverable according to the BIM2TWIN DOW. The lead beneficiary partner should be listed on the front page as Authors and Partner. If not, that would require an explanation.

**Internal Reviewers:** They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically, this review includes identifying typos, identifying syntax and other grammatical errors, altering content, adding or deleting content.

## BIM2TWIN Key Facts

| | |
|---|---|
| Project title | BIM2TWIN: Optimal Construction Management & Production Control |
| Starting date | 01/11/2020 |
| Duration in months | 42 |
| Call (part) identifier | H2020-NMBP-ST-IND-2020-singlestage |
| Topic | LC-EEB-08-2020<br>Digital Building Twins (RIA) |
| Fixed EC Keywords | - |
| Free keywords | Digital Twin; Graph database; BIM; IA; Machine Learning; Image recognition; process optimization; safety improvement |
| Consortium | 17 organizations |

## BIM2TWIN Consortium Partners

| | Partner | Country |
|---|---|---|
| 1 | CSTB : CENTRE SCIENTIFIQUE ET TECHNIQUE DU BATIMENT | FR |
| 2 | TECHNION: ISRAEL INSTITUTE OF TECHNOLOGY | IL |
| 3 | UNIVERSITY OF CAMBRIDGE | UK |
| 4 | TUM: TECHNISCHE UNIVERSITAET MUENCHEN | DE |
| 5 | INRIA : INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE | FR |
| 6 | FIRA GROUP OY | FI |
| 7 | INTSITE LTD | IL |
| 8 | FUNDACION TECNALIA RESEARCH & INNOVATION | ES |
| 9 | ACCIONA CONSTRUCCTION SA | ES |
| 10 | RUHR-UNIVERSITAET BOCHUM | DE |
| 11 | SPADA CONSTRUCTION | FR |
| 12 | UNIVERSITA POLITECNICA DELLE MARCHE | IT |
| 13 | UNISMART – FONDAZIONE UNIVERSITÀ DEGLI STUDI DI PADOVA | IT |
| 14 | ORANGE SA | FR |
| 15 | SIEMENS AKTIENGESELLSCHAFT | DE |
| 16 | IDP INGENIERIA Y ARQUITECTURA IBERIA SL | ES |
| 17 | AARHUS UNIVERSITET | DK |

## DISCLAIMER

## Executive Summary

*BIM2TWIN is conceived with the objective of applying the digital twin concept to the building construction process through a complete and holistic approach. This will be possible thanks to the design and creation of a digital twin platform of the construction process that will have a set of applications that allow the management of the construction and can provide a complete knowledge of the real situation of the work. The BIM2TWIN platform will bring together multiple data sources from the construction site to provide information on the status of the construction process. Knowing what is really happening on site will allow optimizing construction management. The development of the Digital Twin of the construction process on site will focus on the most relevant activities for the construction phase, such as the control of the execution and quality of the works, the planning and control of the necessary equipment and resources, and the safety of the workers.*

*The BIM2TWIN platform aims at collecting low-value data coming form site agents and sensors and delivering high-value data as indicators for decision making. This document is dedicated to the data architecture of the Digital Twin as the way to organize data in a homogeneous and coherent framework. We propose a graph-oriented data model based on web semantic principles.*

*Data is organized in three different layers : Raw Data Layer, Logical Layer and the KPIs Layer. Those layers are modelled by using ontologies.  We have evaluated existing ontologies regarding BIMT2WIN use cases and requirements provided by other workpackages. Some domains were already covered by existing ontologies such as Building Topology or Sensor Networks, others like Worker Safety needed new specific concepts, properties and relations to be designed.*

*The integrated model consists of a Core Model linking Specific-Domain Models together. All concepts of the model including taxonomies are documented through a dictionary approach. An overall workflow describes flow of data through data processors updating the three layers. This workflow considers real-time workflow needed to raise alert for immediate and local decision as well as mid-term workflow for issue reporting and continuous improvement of processes during the construction projects. Eventually, a list of implementation requirement should allow to specify and develop the platform functionalities needed to integrate, populate the model and infer on the data.*

## Table of Contents

## LIST OF TABLES

## List of Figures

## List of Tables

**Aucune entrée de table d'illustration n'a été trouvée.**

## Abbreviations

| | |
|---|---|
| AHRS | Attitude and Heading Reference Systems |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| B2T | BIM2TWIN |
| BIM | Building Information Model |
| BSDD | BuildingSmart Data Dictionary |
| CAD | Computer-Aided Design |
| CDE | Common Data Environment |
| CPM | Critical Path Method |
| DBT | Digital Building Twin |
| DBTP | Digital Building Twin Platform |
| DL | Description Logics |
| DT | Digital Twin |
| ECSO | European Construction Sector Observatory |
| EU | European Union |
| GA | Grant Agreement |
| GPS | Global Positioning System |
| HFM | Heat Flow Meter |
| H&S | Health and Safety |
| IFC | Industry Foundation Classes |
| IoT | Internet of Things |
| IPD | Integrated Project Delivery |
| IQ | Intelligence Quotient |
| ISO | International Organization for Standardization |
| KPI | Key Performance Indicator |
| LBMS | Location-Based Management System |
| LD | Linked Data |
| LIDAR | Laser Imaging Detection and Ranging |
| LPS | Last Planner System |
| MVD | Model View Definition |
| NASA | National Aeronautics and Space Administration |
| ODP | Open Data Platform |
| OHS | Occupational Health and Safety |
| OWL | Web Ontology Language |
| PoC | Proof of Concept |
| PPE | Personal Protective Equipment |
| PtD | Prevention through Design |
| QIRT | Quantitative Infrared Thermography |
| QR | Quick Response |
| R&D | Research and Development |
| RDF | Resource Description Framework |
| RDFS | Resource Description Framework Schema |
| SPARQL | SPARQL Protocol and RDF Query Language |
| SD | Secure Digital |
| SVM | Support Vector Machines |
| SWT | Sementic Web Technologies |

| | |
|---|---|
| URI | Unified Resource Identifier |
| WIFI | Wireless Fidelity |
| WP | Work Package |

# 1 INTRODUCTION

## 1.1 Scope and Objectives

The objective of the BIM2TWIN project is the creation of a Digital Building Twin (DBT) platform to improve efficiency in the management of building construction processes. BIM2TWIN proposes a comprehensive and holistic approach for the application of the Digital Twin concept to building construction: It will enable the involved agents to know the real-time status of everything happening on site and throughout the supply chain: the current progress and quality of the work, the current location of workers, equipment and materials, safety conditions, etc. This will be achieved by conceptualizing, implementing, and demonstrating a Digital Building Twin process.

The Digital Building Twin (hereinafter called DBT) must be able to capture the physical state of the building and the state of the construction process as it is. It must correspond directly to both, the building design, and the defined construction plan, so that everyone involved has access to reliable, accurate, real-time information on the status of the project, essential information for coordinating their work with others. This will ensure that everyone has a thorough understanding of the status of the project. This will require understanding key performance and situational awareness issues during the construction processes, as well as identifying opportunities for improvement in terms of safety, work progress monitoring, optimal resources utilization (including equipment and people) and scheduling.

The main objective of this deliverable is to model the construction site in progress. This model will allow comparing the as-planned and as-performed processes, as well as the as-designed and as-built building elements in terms of quality while monitoring and optimizing impact on worker safety and equipment use. This model will be designed while aligned with existing ontologies of the domain. Some aspects are fully aligned with standard ontologies; other aspects are innovative and specific to the project frame. New aspects will result in a new ontology.

**The model designed in this study is mainly based on the knowledge graph approach that puts the focus on static and logical model, semantic data and inference capabilities.**

Raw data captured on site will be processed and converted into Linked Data before being imported into the platform. Data updates and requests are performed through APIs to: update status to mirror physical site entities, store KPIs calculation results, perform inferences, request KPIs.

## 1.2 Relation to other Work Packages

In WP1 we have defined an ideal construction management process based on a digital building twin. WP1 provides an ideal model and KPIs that will serve as a basis for the information that will be displayed on the control dashboard. The tools developed in WP3 to WP7 aim at capturing real scenarios and gathering current on-site construction practices.

Although the project has an EU-wide coverage, it has been focused on the countries where the demonstrators and pilot cases (WP8) will be carried out. Therefore, from the beginning of the project and specifically in this deliverable, the analysis of the current construction processes has been carried out jointly with the three construction companies in Finland, France, and Spain, where the demonstrators will be conducted.

Task 2.1 is directly connected to task 2.2 by providing the model needed to develop BIM integration tools. It is also connected to Tasks 2.3 to 2.5 to ensure that input data captured on-site and pre-processed can be stored according to the model and with the appropriate level of detail. Task 2.1 is performed jointly with tasks 2.6 & 2.7 to ensure that the platform embeds the whole knowledge of the model and provides required functionalities to import site description and targeted building, update work status and progression and request KPIs to and from the knowledge graph.

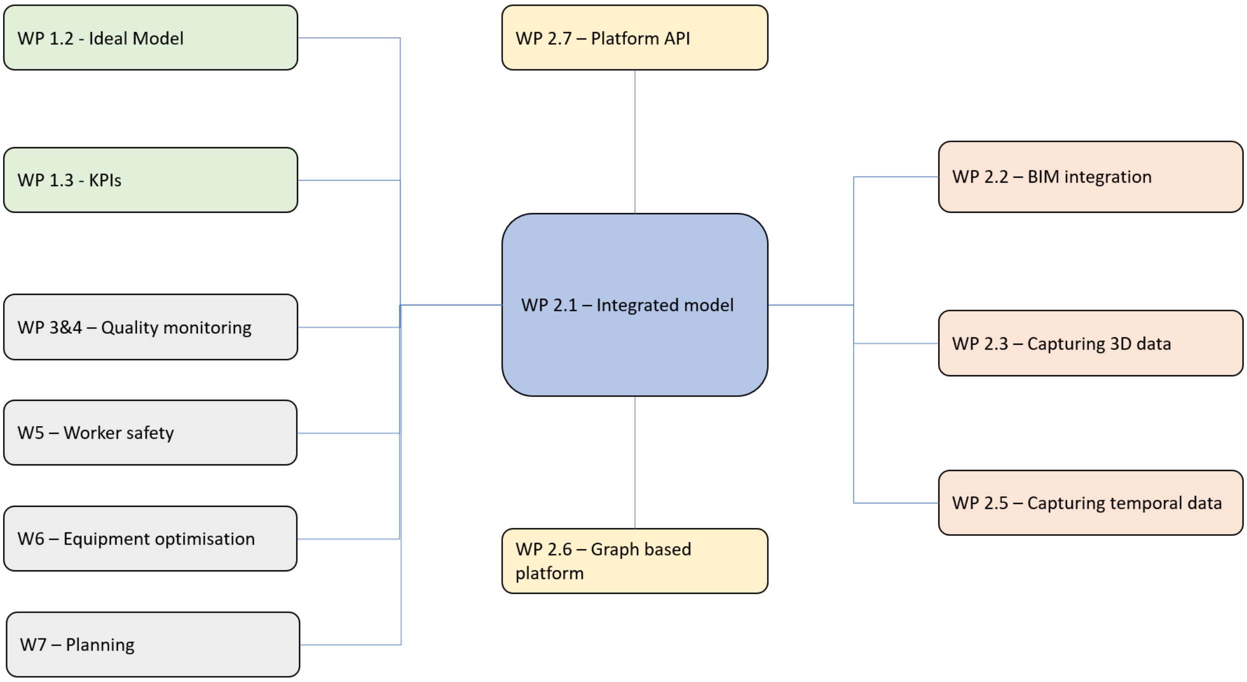


**Figure 1. Project scheme and relationship between work packages**

Practically, links with other work packages and tasks are planned as follows:

WP 1.2: ensure that main concepts are defined in the model

WP 1.3: ensure that KPI are considered in the model

WP3 to WP6: ensure that main concepts are defined in the model so that KPIs can be inferred or calculated from pre-processed data.

WP2.2: ensure that as-perform, as-planned and as-designed BIM model are integrated

WP2.3 & 2.5: ensure that data captured on site can be processed and imported into the platform

WP 2.6 & 2.7: ensure that conceptual model fit with internal meta-model and capabilities of the platform

## 2 CONCEPTUALIZATION OF THE BIM2TWIN DOMAINS WITHIN THE CONSTRUCTION PROCESS

The digital twin concept has been widely adopted by digital manufacturing research and plays a significant role in the industry 4.0. It provides solutions to model, collect, link and process historical and heterogenous data. Concerning construction industry, the digital twin is situated in the meeting point between BIM and IoT. A construction site digital twin implies to model the building but also the process that drive all the construction works. Those process can be compared as those of a manufacture factory with two significant differences: the factory is temporary and dedicated to only one building.

This chapter provides an overview of the construction domains cover by the project. A definition of the B2T domains, on which the BIM2Twin DBT platform will focus is provided in chapter 6, where indicated the scope of each area, defining the performed activities, the involved stakeholders, the relationship with the rest of the construction process, the exchanged information etc.

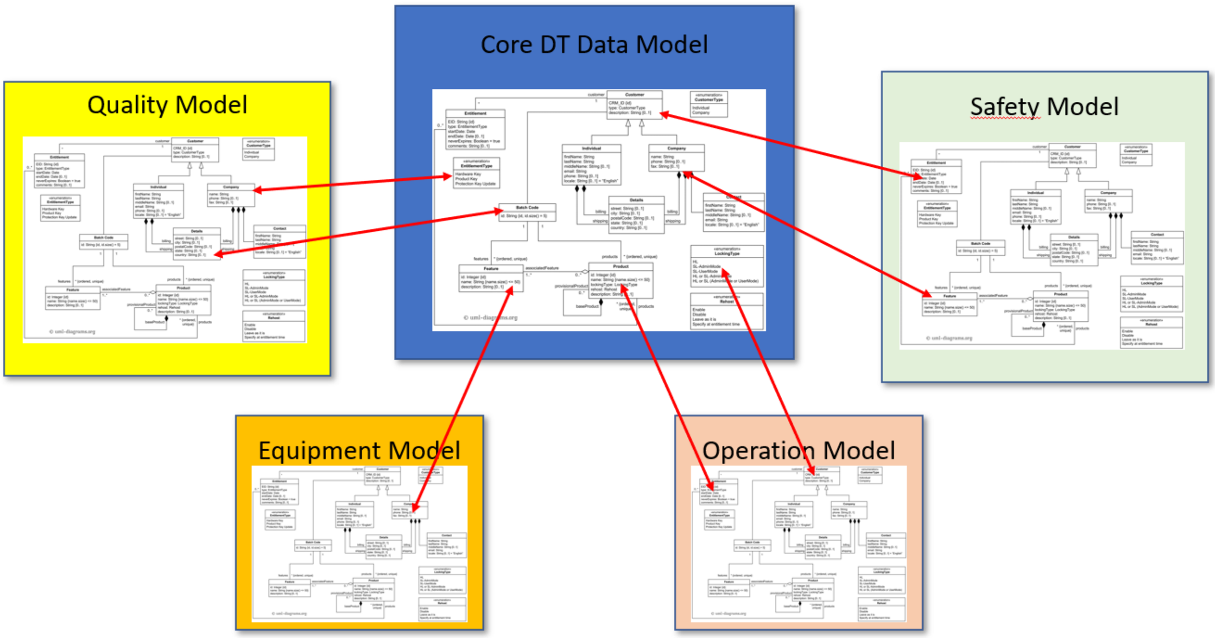


**Figure 2 - BIM2TWIN domains overview**

The BIM2TWIN project aims to improve efficiency of the construction process management by focusing on the on-site work phase. To achieve the optimization of the construction phase, the project developments will focus on 3 specific domains tackled on the B2T project which are: Quality control of as-built element (Quality Model), On-site Worker Safety (Safety models) and equipment and Production planning (equipment and Operation Model). All those model model are connected through a Core Data Model including main and shared concepts.

## 3 RESEARCH METHODOLOGY

The construction site digital twin data models organize data that allow to mirror real construction projects. In the current project we mainly focus on the main relevant use cases studied in WP3-7 (quality, construction processes productivity, equipment, safety). The data schema is organized in three different layers (described in Chapter 7) from the raw data layers (data coming from sensors and users) to the KPI layers (main performance indicators), including a logical layer in between. We have identified two kinds of data processes that need to be considered separately. On one hand, real time processes detect instant and local issues directly from raw data. On the other hand, mid term processes, based on the logical layer, keep trace of all events with time step from day to week. It allows to calculate KPIs, and make strategic decisions. In this document we mainly focus on logical layer and midterm processes but provide connections with lower and uppers layers and real time processes.

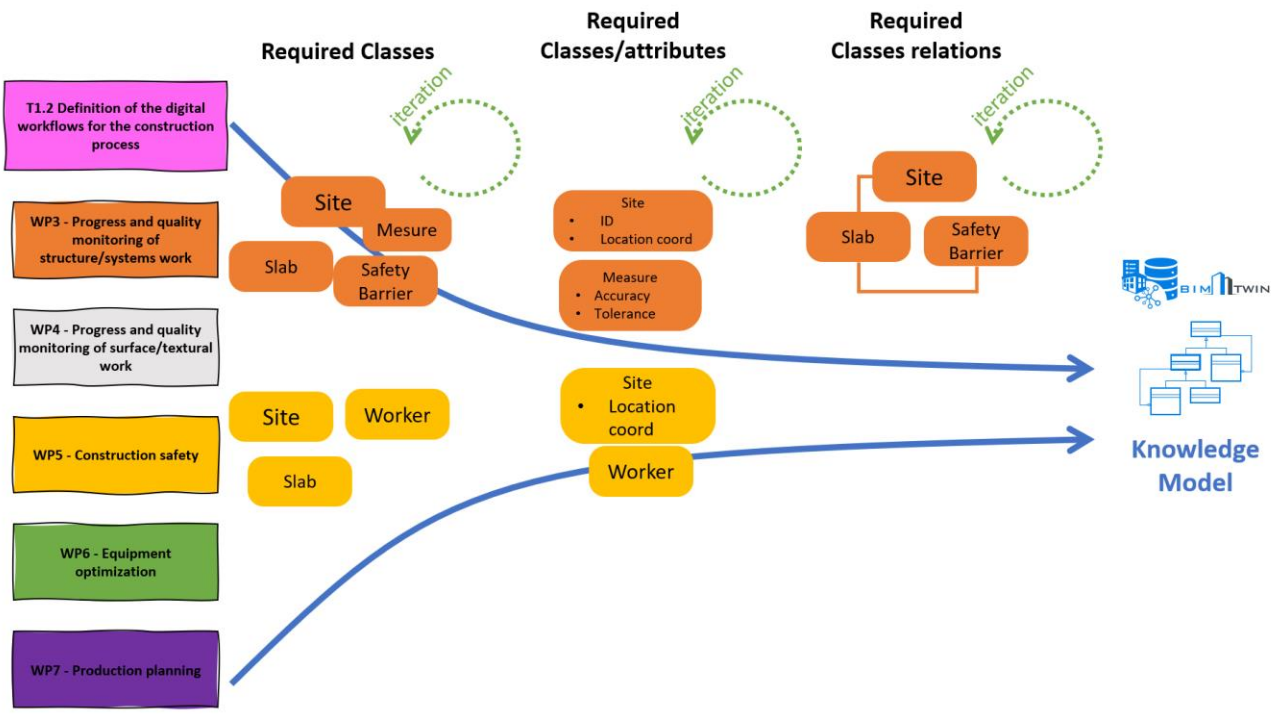


**Figure 3 - From on-site use cases to knowledge model**

Our approach consists of designing a knowledge graph by analysing use cases from work packages WP1 and WP3 to 7 to find the main logical classes, enrich them with attributes and taxonomies and then derive a the model. The core part of the model contains the classes that are necessary independent of the present use case. It will be extended through classes that are specific to the use cases in WP3 to 7.

## 3.1 Top-down approach

The digital twin aims at helping the decision makers to keep the construction project on track. WP1 T1.2 defines the ideal and innovative construction processes as a target for construction processes improvements. To derive technical requirements, a formal information analysis has been conducted to analyse the user requirements defined in the first phase of WP1 T1.2. The T1.2 output has been considered to identify the main concept of the construction site digital twin. WP1 T1.3 defines the KPIs that drive the logical layer outputs.

The main KPIs are the following:

- For process optimisation: Throughput, cycle time, order fulfilment, work capacity, workforce utilization rate and equipment utilization rate.
- For quality monitoring: Number of detects identified/fixed per work package, number of executions with no detects, delays due to quality.
- For safety control: site safety level, defect solving times, risk assessment completion level, accident frequency

## 3.2 Bottom-up approach

We have considered the system design documents generated by the individual digital twin services (WP3-7) to find workflow paths from available raw data to needed KPIs. When available data is not sufficient to get an accurate measurement of reality and approximate processing is proposed to be able to at least raise warnings and alerts. Those warnings inform the decision makers what kind of defect need to be fixed or prevented. The bottom-up approach focuses on available data and drives logical layer inputs.

# 4 META MODEL

Before beginning model analysis, the first issue is the choice of the most adapted meta-model[1].

“A metamodel or surrogate model is a model of a model, and metamodeling is the process of generating such metamodels. Thus, metamodeling or meta-modelling is the analysis, construction and development of the frames, rules, constraints, models, and theories applicable and useful for modelling a predefined class of problems. As its name implies, this concept applies the notions of meta- and modelling in software engineering and systems engineering. Metamodels are of many types and have diverse applications.” [2]

In 2014, the Object Management Group (OMG) introduced the Meta-Object Facility (MOF)[3] as a standard for model-driven engineering. MOF is designed as a four-layered architecture. It provides a meta-meta model at the top layer (M3 layer). This M3-model is the language used by MOF to build metamodels, called M2-models. These M2-models describe elements of the M1-layer, and thus M1-models. The last layer is the M0-layer or data layer. It is used to describe real-world objects.

The meta model should conform to the followings requirements.

- Provide interoperability with industry practices and open standards
- Gather classes, attributes, properties and taxonomies in the same model
- Provide an embed documentation with unambiguous definition of concepts
- Extensibility

As Shen Dai and al.[4] showed, compared with information methods using other languages such as Unified Markup Language (UML)[5], Extended Markup language (XML)[6] or EXPRESS[7], ontologies[8] provide a rich description vocabulary with better machine-readable capability, which can define complex relationships.

The vocabulary, involved in BIM2TWIN scope, concerns all types of abstract and physical entities that exist on a construction site and during a construction process. This vocabulary is the first material for and evaluating existing model and modelling specific domains.

The Semantic Web allows to organize this domain vocabulary through a Knowledge Graph. It provides a MOF based meta-model[9] with the four layers : OWL–Web Ontology Language (M3), General Ontology (M2), Domain specific Ontology (M1), RDF resources (M0).

## 4.1 The Semantic Web approach

When a model is tied to a particular industry, it can be easier and more effective to use a set of already existing models for that industry as a starting point, instead of authoring a new model set from scratch. In the context of the Semantic Web[10] and Linked Data[11] these pre-existing model sets are called ontologies. An ontology is an explicit specification of concepts that provides a comprehensive specification of knowledge for a domain. It provides a basis for semantic relation mining for systems such as ours that process information and infer knowledge[12]. In general, an ontology is a set of models for a given domain—like a building structure, IoT system, smart city, the energy grid, web content, etc. Ontologies are often used as schemas for twin graphs, as they can enable:

- Harmonization of software components, documentation, query libraries, etc.
- Reduced investment in conceptual and system development

- Easier data interoperability on a semantic level
- Best practice reuse, rather than starting from scratch or "reinventing the wheel"

In order to overcome current hurdles described above, we propose the application of Semantic Web technologies (SWT). Semantic Web provides a powerful toolset to define and maintain a controlled vocabulary of processes, roles, objects and interactions. The Semantic Web expands on the current World Wide Web (WWW) framework. Linked Open Data allows for data to be read and interpreted by both humans and machines, consequently better enabling cooperation between computers and humans. While the traditional WWW links information via human-readable documents encoded in Hypertext Markup Language (HTML)[13], Semantic Web links information on the data level using the Resource Description Framework (RDF)[14]. Therefore, it is machine-understandable and -interpretable and P. Moder et al.[15] hence improves data analysis possibilities as well as knowledge extraction[16].

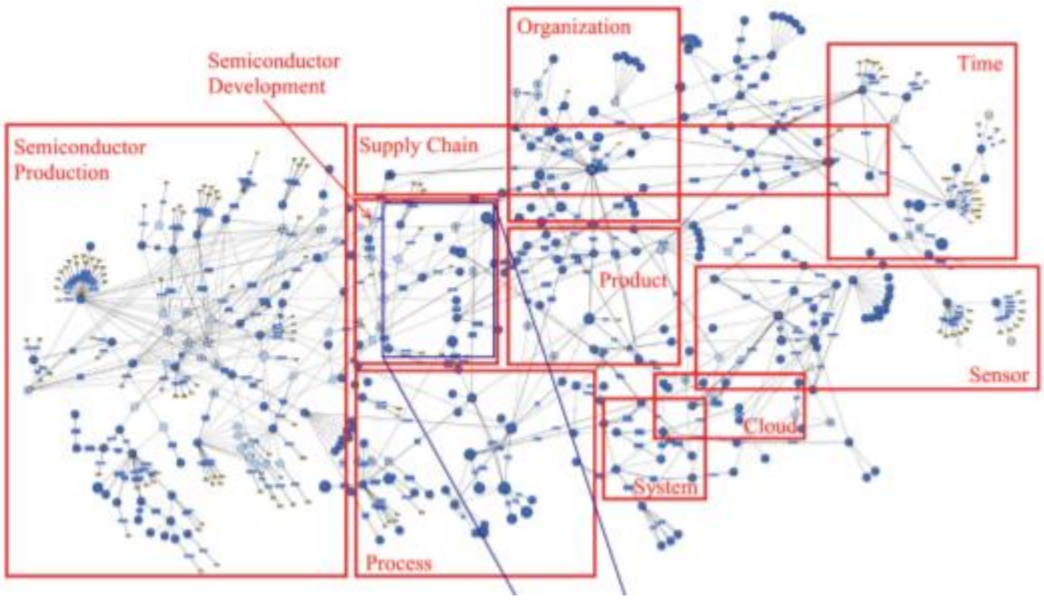


**Figure 4 - Digital Twin based on semantic web and applied to Semiconductors**

First introduced in 2001 by Tim Berners-Lee[17], the dual intelligibility between humans and machines provides a tremendous opportunity as an enabler for emerging technologies such as blockchain, Big Data analytics and automation. Semantic Web is also a big step towards advancing several of the design principles of digitalization and Industry 4.0[18].

Ontology is the named given to the model for Graph oriented database while Relational Databases use Merise[19] Models.

In the early stages of graph modelling through ontologies is close to the relational database modelling: identifying main concepts and main relations between those concepts. But instead of mapping the resulting model onto relations (tables and fields), the next step is to enrich the already graph-like structure. This way, graph modelling is closer to Meta-Object Facility approach than relational database modelling.

According to Mathias Bonduel[20], after considering different approaches, "Although the adoption of Semantic Web Technologies (SWT) for the description of buildings in practice is still limited, more and more researchers investigate the opportunities of this technology stack. The outcomes of their work suggest that SWT might be beneficial to stakeholders for the management of construction projects."

According to Zhu et al.[21], ontologies have been widely used in context modelling, as they are independent of programming languages and enable context reasoning. Stark et al.[22] use ontologies in the context of PLM to improve sustainable product development. Abramovici et al.[23] offer a semantic data management platform for the development and continuous reconfiguration of smart products and systems. There is also a rich literature in computational ontologies, which cover Description Logic used broadly for defining ontologies such as the Web Ontology Language[24].

Ontologies provides a great starting point for a digital twin solution. They encompass a set of domain-specific models and relationships between entities for designing, creating, and parsing a digital twin graph. Ontologies enable solution developers to begin a digital twin solution from a proven starting point and focus on solving business problems.

Those reasons partially explain why digital twin research has turned to ontologies as meta-models. Digital Twin platforms have accelerated to adaption of graphs by various industries. The digital twin as the meeting point between heavy building processes and IOT (Internet of Things)[25] forces us to rethink data models traditionally used in Architecture and Construction Industry (AEC). This is the perfect time to adopt the knowledge graph[26] as a way of modelling the real world without sacrifice the user point of view.

The Semantic Web is an extension of the World Wide Web through standards set by the World Wide Web Consortium (W3C). The goal of the Semantic Web is to make data on the Internet machine-readable. To enable the encoding of semantics with the data, domain specific ontologies and based on the fundamental ontology language (MOF M3 layer) called Web Ontology Language (OWL)[27]. This language is used to formally represent concepts, relationships between entities and categories of things. These embedded semantics offer significant advantages such as reasoning over data and operating with heterogeneous data sources[28].

To derive a Construction Site Digital Twin, we have followed the four main semantic web principles[29]:

- Use URIs to identify things
- Use HTTP URIs so that people can look up those names
- When someone looks up a URI, provide useful documentation, using the standards (RDF, RDFS, SPARQL[30])
- Include links to other URIs, so that they can discover more things

## 4.2 Reusing standard models

Another main principle of Web Semantic approach is to reuse commonly shared models and aggregate models to fit to domain needs. Firstly, we considered existing ontologies based on standards that fit the construction domain and might be relevant for a Digital Building Twin to keep the site construction model close to standards and avoid reinventing models. The following figure gives a small insight into the sheer number of already existing ontologies one can build upon, separated by the field of application.

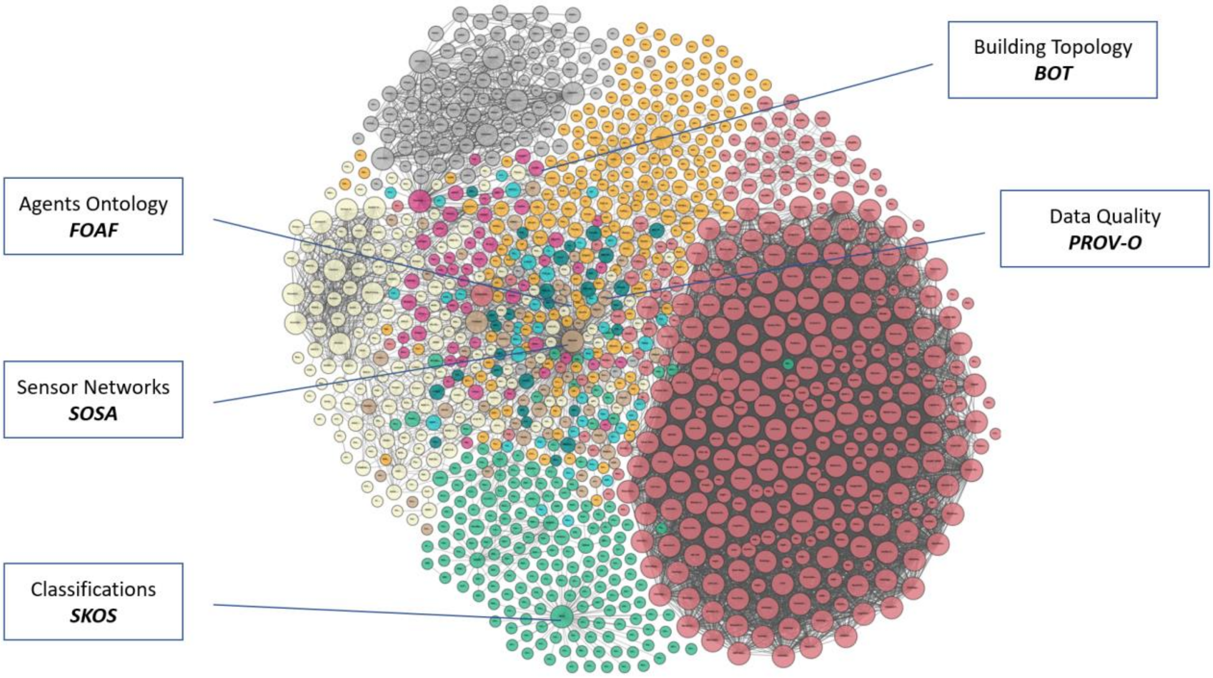


**Figure 5 - Map of existing public ontologies**

There are three possible strategies for integrating industry-standard ontologies for a specific domain like a digital twin for the construction site: federate, align or customize. These three strategies can be mixed and the strategy choice can concern only a part of the whole digital twin model.

**Federate**: Ontology that has been built on widely adopted industry standards and the cover considered domain and use case should be adopted. When an ontology is adapted, new concepts can be directly linked to it.

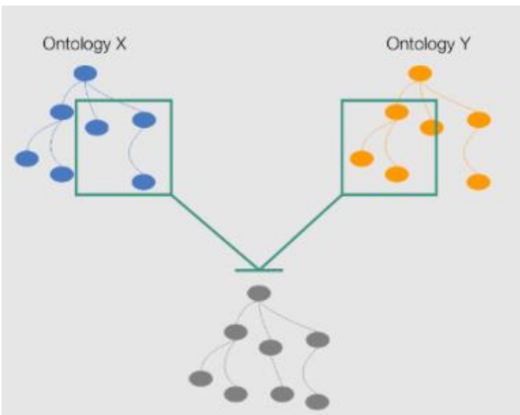


**Figure 6 - Ontology federation**

**Align**: If an already existing model fits with the considered domain but uses a different approach. A specific existing ontology can be aligned totally or partially. Ontology alignment, or ontology matching, is the process of determining correspondences between concepts in ontologies. When schemas are not defined with OWL formalism, it can be converted into OWL before the alignment. For instance ifcOWL is the OWL conversion of the IFC/STEP schema.

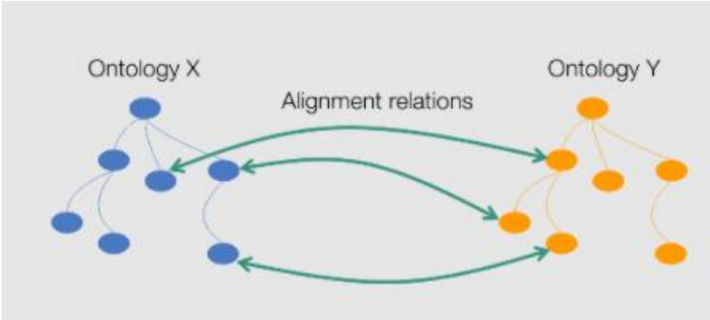


**Figure 7 - Ontology alignment**

**Customize:** When a part of the domain is really specific and is not yet covered by an existing ontology, a custom ontology can be developed using standards as guideline. A new domain ontology needs at least a high level ontology that links to the federated ontology.

After exploring existing ontologies in part 5 and domain model requirements in part 6, we will (in part 7) present our alignment strategy and build the BIM2TWIN ontology by connecting domain model concepts with and federating ontologies around the core model.

## 4.3 Implementing Graph oriented models

Two ways of implementing Graph oriented models exists :

**Property Graphs** : While there are core commonalities in property graph implementations, there is no true standard property graph data model. Each implementation of a Property Graph is, therefore, somewhat different. The most well-known implementation, which popularized Property Graphs as a concept, is the Neo4J graph database. Generally, the property graph data model consists of three elements: Nodes (or entities), Edges (or relationships between entities), and Properties (or node attributes). Both, entities, and relationships can have attributes. There are various languages to query Property graphs. Neo4J offers Cypher as a query language while TigerGraph offers GSQL and Apache TinkerPop offers the Gremlin language. ArangoDB the property graph engine embedded in Thing'in platform implements Gremlin and a JSON-based query language.

**RDF Knowledge Graphs** use a standard graph data model. The standard for the RDF technology stack is managed by the World Wide Web consortium (W3C), the same standards body that manages HTML, XML, and many other web standards. Every database that supports RDF is expected to support the model in the same way. The RDF graph data model basically consists of two elements: Nodes can be resources with unique identifiers, or they can be "literals" with values that are strings, integers, etc. Edges are also called predicates and/or properties. Two nodes connected by an edge form a subject-predicate-object statement, also known as a Triple or a Triple Statement. While edges are directed, they can be navigated and queried in either direction. Everything in an RDF graph is called a resource. "Edge" and "Node" are just the roles played by a resource in a given statement. Fundamentally in RDF, there is no difference between resources playing an edge role and resources playing a node role. An edge in one statement that can be a node in another. There is a standard query language for RDF Graphs called SPARQL. It is both, a full featured query language and an

HTTP protocol making it possible to send query requests to endpoints over HTTP. A key part of the RDF standard is the definition of serializations. The most commonly used serialization format is called Turtle. There is also a JSON serialization called JSON-LD as well as an XML serialization. All RDF databases are able to export and import graph content in standard serializations making it easy and seamless to interchange data. A large of Triple Store implement RDF and SPARQL such as Stardog, GraphDB, Jena or Virtuoso.

**Table 1 - Comparison between RDF and Property graphs**

| **Feature** | **RDF** | **Property Graph** |
|---|---|---|
| **Expressivity** | Arbitrary complex descriptions via links to other nodes; no properties on edges<br>With RDF* the model gets much more expressive than PG | Limited expressivity, beyond the basic directed cyclic labelled graph<br>Properties (key-value pairs) for nodes and edges balance between complexity and utility |
| **Formal semantics** | Yes, standard schema and model semantics foster data reuse and inference | No formal model representation |
| **Standardization** | Driven by W3C working groups and standardization processes | Different competing vendors |
| **Query language** | SPARQL specifications: Query Language, Updates, Federation, Protocol (end-point) | Cypher, PGQL, GCore, GQL (no standard) |
| **Serialization format** | Multiple serialization formats | No serialization format |
| **Schema language** | RDFS, OWL, Shapes | None |

RDF Knowledge Graphs make it easier to reuse and share common and standard models. While Property graph is said to be easier to learn. Those two approaches seem to converge on the long term. RDF as recently adapted RDF* to allow properties on edges. Some Property Graph engines now accept RDF as import format. Some recent Graph databases like AWS Neptune can now handle both kind of graphs.

## 5 LITERATURE REVIEW AND EXISTING ONTOLOGIES, TAXONOMIES AND DATA SPECIFICATION STANDARDS

This part is an overview of the state-of-the-art concerning ontology-based modelling in the context of digital twins and construction processes. Though the following models are not always dedicated to construction site processes, most of the provided concepts can be reused, aligned or specialized for the present digital twin use case. Most of the provided ontologies that belong to this study follow semantic web principles defined by W3C (OWL or RDF schemas). At the implementation level, concept or partial ontology reuse take the form of an *import*. Specialization is made by the use of the RDFS predicate *owl:subClassOf* . Eventually alignment between concepts of different ontologies can be strong with the use of the *owl:sameAs* predicate or loosely with the use of *skos:related* predicate.

*A synthetic table of ontologies presented hereafter is provided in §5.4. Benefits from these ontologies for BIM2TWIN project are discussed in §8.3.*

### 5.1 European directives on data interoperability

**INSPIRE Directive**[31] aims to create a European Union spatial data infrastructure for the purposes of EU environmental policies and policies or activities which may have an impact on the environment. This European Spatial Data Infrastructure will enable sharing of environmental spatial information among public sector organisations, facilitate public access to spatial information across Europe and assist in policy-making across borders. INSPIRE defines geospatial data models and directives at a large scale. Those directives can be considered as soon as it is relevant to geolocate resources on a scale wider than building scale. Inspire defines buildings at land scale, cadastral parcels, and land cover.

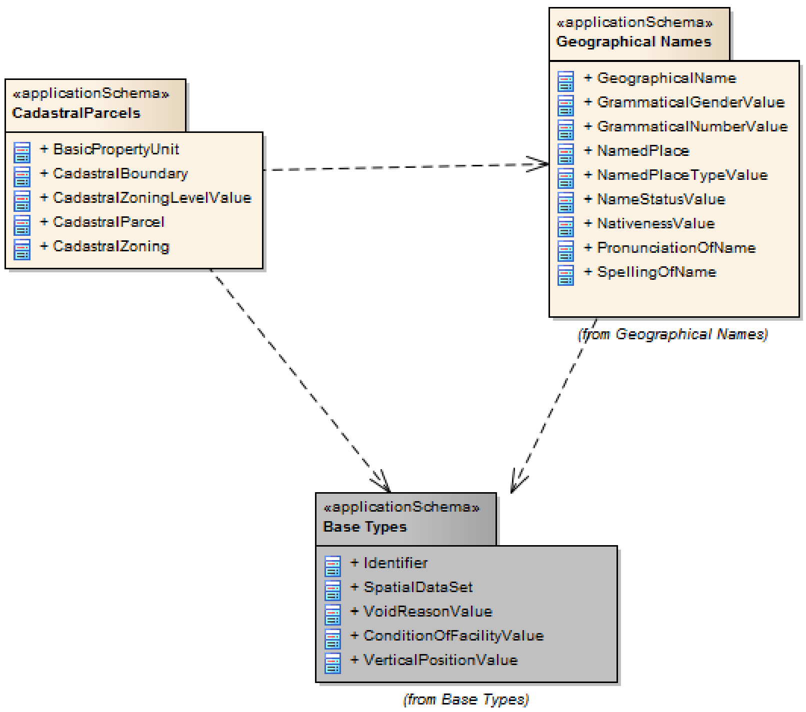


**Figure 8 - Parcels models form INSPIRE set of specifications**

The data specifications provided by INSPIRE are covering urban and territory scales. Those models can be used either for very large construction sites, for the location of the site within a larger urban environment, or for the description of the surrounding urban elements linked with the construction site like hydrography or transport networks. Though in an exhaustive approach of the construction site digital twin those models can be relevant, but they appear to be out of BIM2TWIN scope.

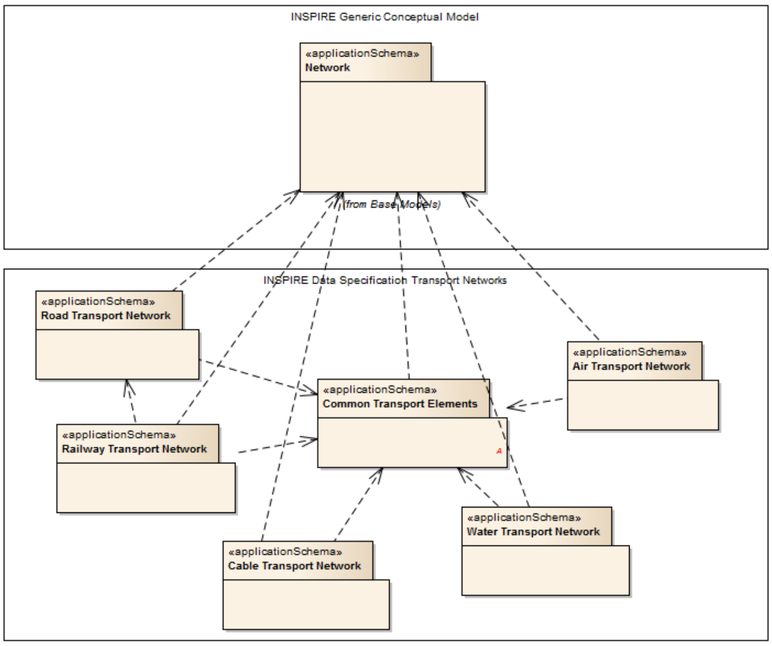


**Figure 9 - Parcels models form INSPIRE set of specifications**

## 5.2 Related European projects

**SPHERE project** is a 4-year Horizon 2020 project that aims to provide a BIM-based Digital Twin Platform to optimise the building lifecycle, reduce costs, and improve the energy efficiency of residential buildings.

The following table compile a inventory of ontology candidate for Digital Building Twin. Each ontology is evaluated against various criteria such as extensibility, semantic richness, compatibility with IFC, document quality, etc.

**Table 2 - Sphere benchmark on existing ontologies covering Building Digital Twin scope**

| | A | C | D | E | F | G | H | I | J | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | MUST HAVE | | | | |
| 2 | Ontology | Last update | Estimated installed base | Proven compatibility | Domain independence | Extensible | Semantic richness | Sensor resources | Sensor data | Compatibility IFC | Detailed description | No datatype limitation | Availability/License |
| 3 | SSN | 2017 | large | DogOnt, IoT-O, BACS, | OK | OK | OK | OK | OK | OK | OK | OK | OK |
| 4 | Xue et al | 2015 | | | NOK | OK | ? | OK | OK | | | | NOK |
| 5 | M3 | 2020 | small | IFC, SSN | OK | OK | OK | OK | OK | OK | | | NOK |
| 6 | OntoSensor | 2008 | | | OK | NOK | | OK | OK | | | | |
| 7 | MyOntoSens | 2015 | very small | SSN | OK | | ? | OK | | | ? | | NOK |
| 8 | Proposed by Hirmer et al. (based on SensorML) | | | | | | | OK | OK | | | | |
| 9 | Sensor Core Ontology (Shi et al) | 2012 | very small | | OK | OK | | OK | OK | | NOK | | |
| 10 | SAREF | 2018 | ? | IFC | NOK | OK | OK | OK | NOK | OK | OK | | |
| 11 | SAREF4EE | | | IFC | NOK | OK | | OK | OK | OK | | | |
| 12 | SAREF4BLDG | | | IFC, SSN | NOK | OK | | OK | OK | OK | | | |
| 13 | Ontology by Dey et al. for smart energy meters | 2014 | low | | NOK | | | | | | | | |
| 14 | BRICK | | | IFC, RealEstateCore (v | NOK | | NOK | OK | | OK | NOK | | OK |
| 15 | IoT-O | 2018 | | SSN | | NOK | | OK | OK | | NOK | | |
| 16 | BACS Ontology | 2017 | small | SSN | NOK | | OK | OK | OK | OK | depends on part | OK | OK |
| 17 | DogOnt | 2019 | | SSN | NOK | OK | | OK | OK | | | | |
| 18 | ifcOWL | | | SSN | OK | OK | OK | NOK | NOK | OK | OK | NOK | |
| 19 | BONSAI | | large | IFC | NOK | OK | NOK | NOK | NOK | OK | NOK | | |
| 20 | COSE ontology: Casas Ontology for Smart Environments | 2011 | very small | | | | | | | | | | NOK |
| 21 | SBOnto | 2017 | | | NOK | | NOK | | | | NOK | | NOK |
| 22 | oneM2M | | large in industry | | | | | | | | | | docx |
| 23 | Haystack tagging ontology | | | | NOK | | NOK | | | | OK | | |
| 24 | OEMA ( ontology for energy management applications) | | | | NOK | | | | | | | | |
| 25 | BASOnt | | | IFC | NOK | | | | limited | OK | NOK | | paper |
| 26 | ASHRAE (BACNet) | | | | OK | | | NOK | NOK | | | | OK |
| 27 | RealEstateCore | 2019 | | IFC (v4), SSN (v4), BR | NOK | OK | OK | OK | OK | v4 OK | OK | OK | OK |
| 28 | gbXML | 2017 | | | NOK | | | OK | OK | | NOK | | |
| 29 | IMDF | | | | NOK | | | | | | | | |
| 30 | | | | | | | | | | | | | |

Sphere provides a state-of-the-art of ontologies concerning Building Digital Twins[32]. As BIM2TWIN is rather focused on the construction processes optimization than on the building performance during the operating phase, only a part of the ontologies that are used will be shared between the two projects such as the IOT model (SSN, SAREF) and the as-design building models (BOT).

**Linked vocabulary for BIM[33] :** this study is an overall comparison among the relevant vocabularies to help to decided on ontologies to use in future BIM applications based on Linked Data. It provides criteria to evaluate vocabularies.

The main ontologies from those studies that appear to fit with in the scope of BIM2TWIN are explained in detail in the following paragraphs.

**BIMERR project[34]** is related to Building Information Modelling (BIM) and its main target are stakeholders from the AEC (Architecture, Engineering & Construction) field. The project has the intention to design and develop a new toolkit to support stakeholders during the renovation process of existing buildings, from concept to delivery. Though BIMERR is mainly dedicated to renovation, it identifies and provides low level concepts that can be reused for site construction modelling.

**BIM4EEB**[35] is an Horizon 2020 project that develops a BIM-based toolset able to support designers in the design and planning phase, construction companies to efficiently carry out the work and service companies to provide attractive solutions for building retrofitting. BIM4EEB is on of the contributing projects to **Digital Construction Ontologies (DiCon)** detailed in the following state-of-the-art that.

## 5.3 Ontologies covering Data quality domain

The **PROV Ontology**[36] (PROV-O) expresses the PROV Data Model [PROV-DM] using the OWL2 Web Ontology Language (OWL2) [OWL2-OVERVIEW]. It provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. It can also be specialized to create new classes and properties to model provenance information for different applications and domains. The PROV Document Overview describes the overall state of PROV and should be read before other PROV documents.

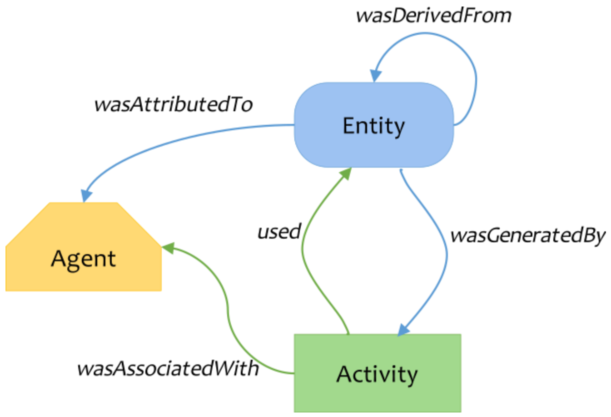


**Figure 10 - PROV-O core concepts**

**The Key Performance Indicator ontology**[37] aims to model the metrics defined at the beginning of building renovation activities to monitor the conformance with typical requirements related to energy efficient buildings.

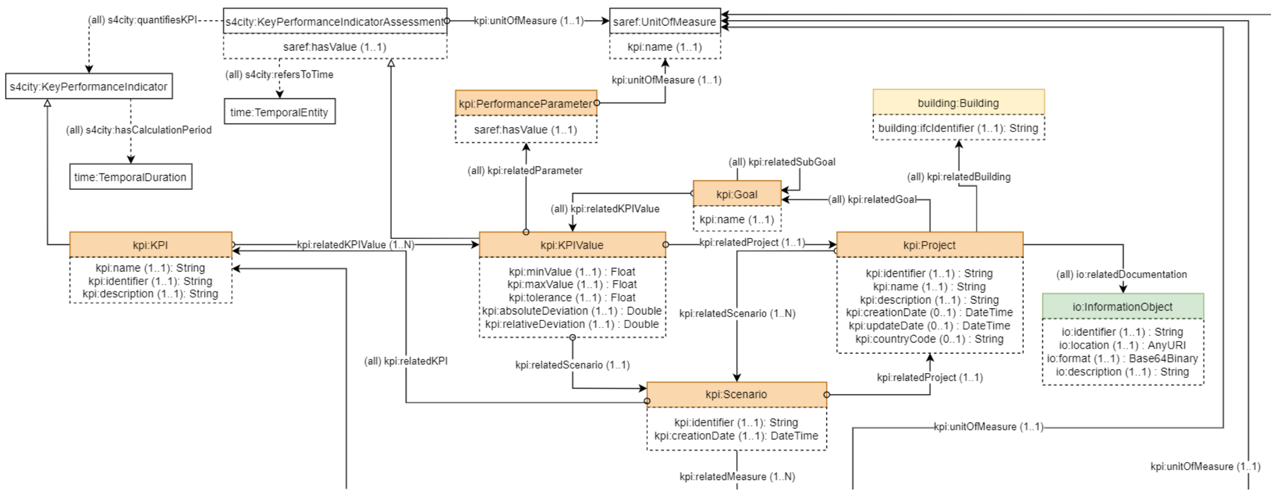


**Figure 11 - Key Performance Indicator Ontolgoy**

**QUDT**[38] is a reference ontology for most of the engineering models, when physical measurement and Unit are needed. It provides a set of ontologies with a unified model of, quantities, their dimensions and units well as instances data with particular quantities, units and values. In QUDT, a Quantity is an observable property of an object, event or system that can be measured and quantified numerically. **Quantity Kind** is any observable property that can be measured and quantified numerically. **Unit** is a particular quantity of a given kind that has been chosen as a scale for measuring. Eventually, a **Quantity Value** expresses the numerical value of a quantity with respect to a chosen unit of measure.

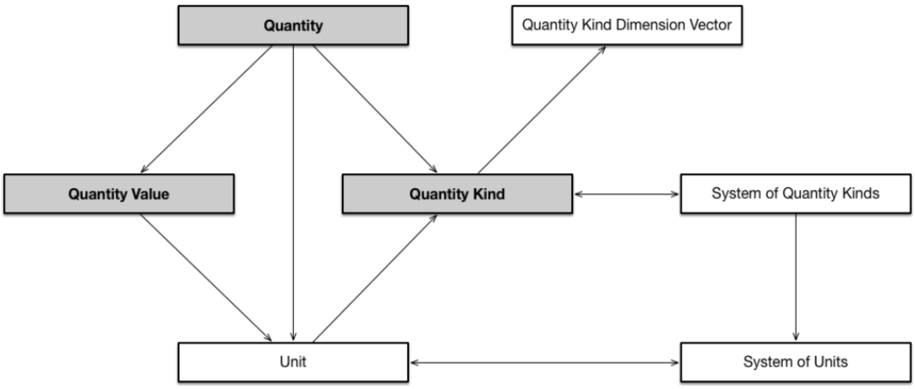


**Figure 12 - QUDT Ontology**

QUDT can be used to implements the use cases itemized below:

- Conversion between single and complex unit types
- Dimensional analysis of equations
- Finding out equivalent units in different systems of units
- Finding out equivalent quantity kinds in different systems of quantities

## 5.4 Ontologies covering Terminology and Taxonomy domain

The **Simple Knowledge Organization System (SKOS)**[39] is a common data model for sharing and linking knowledge organization systems via the Semantic Web. This document provides a brief description of the SKOS RDF Schema.

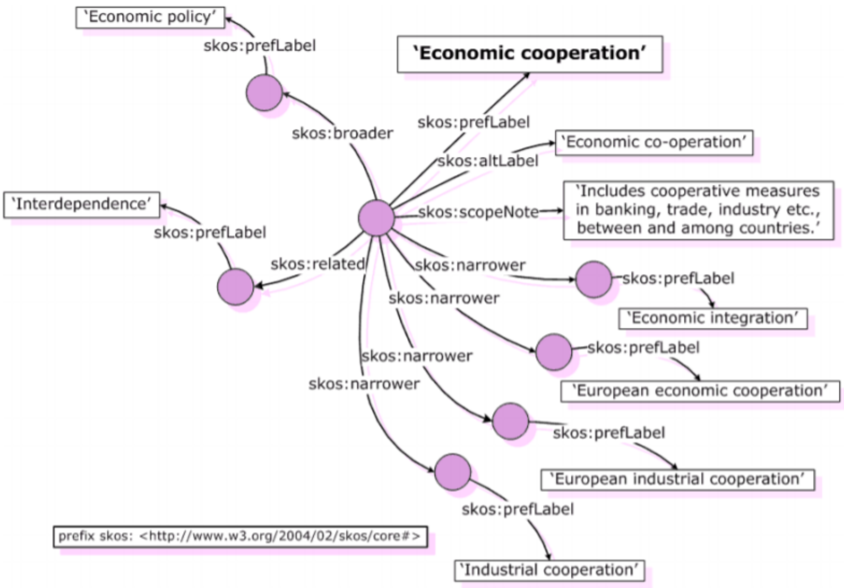


**Figure 13 - SKOS Ontology**

The SKOS ontology is used to define vocabulary in early stage of modelling works. The BIM2TWIN dictionary is based on SKOS to define vocabulary terms and semantic links between terms.

**ISO 12006-3 :2007 "Building construction —Organization of information about construction works — Part 3: Framework for object-oriented information".** ISO 12006-2:2015 defines a framework for the development of a built environment classification systems. It identifies a set of recommended classification table titles for a range of information object classes according to particular views, e.g., by form or function, supported by definitions. It shows how the object classes classified in each table are related, as a series of systems and sub-systems, e.g., in a building information model. bSDD is one implementation of ISO 12006 promoted by BuildingSmart. It gives access to building concept definitions and their attributes through an online service. bSDD promotes IFC4 taxonomies a property sets and the Omniclass classification.

**Table 3 - Subsection of the Omniclass Elements classification**

| OmniClass Number | Level 1 Title | Level 2 Title | Level 3 Title | Level 4 Title | Table 22 Reference |
|---|---|---|---|---|---|
| 21-01 00 00 | **Substructure** | | | | |
| 21-01 10 | | **Foundations** | | | |
| 21-01 10 10 | | | **Standard Foundations** | | |
| 21-01 10 10 10 | | | | Wall Foundations | |
| 21-01 10 10 30 | | | | Column Foundations | |
| 21-01 10 10 90 | | | | Standard Foundation Supplementary Components | |
| 21-01 10 20 | | | **Special Foundations** | | 22-31 60 00 |
| 21-01 10 20 10 | | | | Driven Piles | 22-31 62 00 |
| 21-01 10 20 15 | | | | Bored Piles | 22-31 63 00 |
| 21-01 10 20 20 | | | | Caissons | 22-31 64 00 |
| 21-01 10 20 30 | | | | Special Foundation Walls | 22-31 66 16 |
| 21-01 10 20 40 | | | | Foundation Anchors | 22-31 68 00 |
| 21-01 10 20 50 | | | | Underpinning | 22-31 48 00 |
| 21-01 10 20 60 | | | | Raft Foundations | 22-03 71 00 |
| 21-01 10 20 70 | | | | Pile Caps | |
| 21-01 10 20 80 | | | | Grade Beams | |
| 21-01 20 | | **Subgrade Enclosures** | | | |

As a classification repository bSDD also gathers other several system and product taxonomies specialized by domain or country of application. Some of the classifications are overlapping each other making it good tools for taxonomy alignments. As BEO, products taxonomies like OmniClass or Uniformat can enrich Building Model's native taxonomy (IFC[40]). Concerning properties, the IFC4 level of information seems to be sufficient to cover BIM2TWIN scope for the building description as-designed.

The **ConTax ontology**[41] contains additional terminology for structuring and annotating RDFS/OWL taxonomies for describing constructions (components, materials, spatial zones, damages, construction tasks and properties). It also functions as an index for known taxonomies starting from root classes and properties.

The **BEO (Building Element Ontology)**[42] can be considered as one of the ontologies that are based on a portion of IFCOWL. BEO is an ontology that only represents the part of *IfcBuildingElement* with all of its subclasses. The BEO ontology provides to IFC a extension of the products taxonomy. BEO taxonomy will be integrated in the BIM2TWIN products taxonomy.

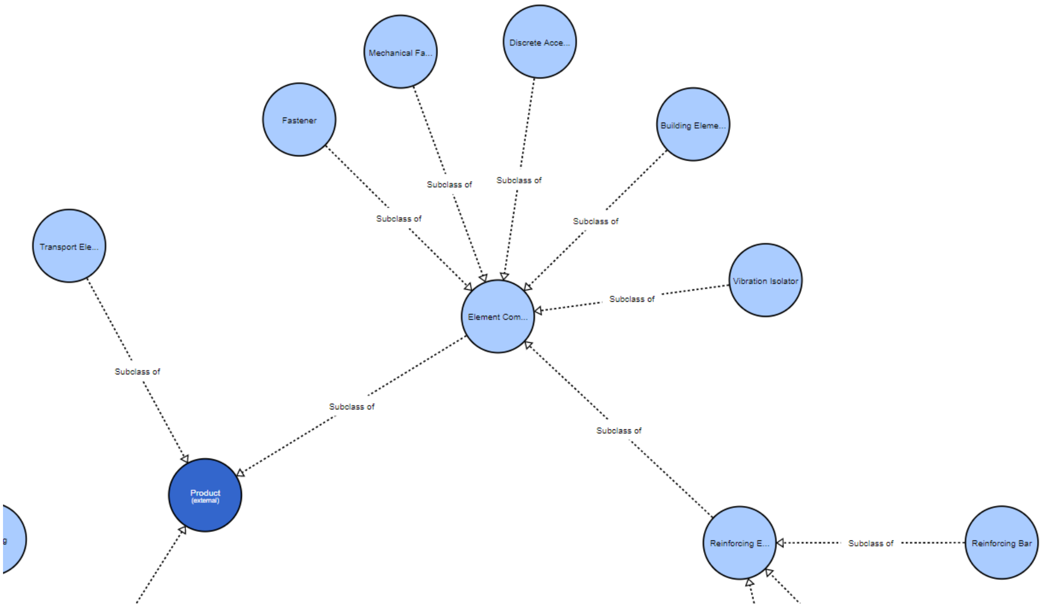


**Figure 14 - Subsection of the Products Taxonomy provided by BEO**

Various existing taxonomies are extracted and restructured by the Getty Art & Architecture Thesaurus (Getty AAT), v2020-07-29. **AAT-ARCH taxonomy**[43] contains OWL classes to classify architectural construction (sub)components, extending respectively contax:ConstructionComponent and contax:ConstructionSubComponent. **AAT-FURN taxonomy**[44] contains OWL classes to classify furniture components, extending contax:ConstructionComponent. **The AAT-MEP taxonomy**[45] contains OWL classes to classify MEP construction components, extending contax:ConstructionComponent.

### 5.5 Ontologies covering geometric descriptions

The geospatial data provided by many Linked Open datasets including GeoNames (and DBPedia by proxy) uses the **WGS84 ontology**[46]. WGS84 is a minimalistic RDF vocabulary. The intention of the ontology is not to address many of the issues associated with representing geometries and locations; it is solely to provide basic terms in RDF to describe point locations. There are many obvious limitations to representing data using the WGS84 ontology, including representing other shapes such as polygons and linestrings, and representing data in other coordinate systems.

Simplified geometry (for low Level of Detail purposes) can be described by using **Well Known Text Coordinates Representation (WKT)**[47] as literals. A well-known and OGC-W3C standardized implementation is **GeoSPARQL**[48], which supports 2D shapes using WKT or **Geography Markup Language (GML)**[49]. Besides a set of linking terminology, the standard defines an extension of SPARQL for 2D geospatial queries. BimSPARQL, as proposed in academic literature[50], is similar to GeoSPARQL but is targeted at 3D geospatial querying over 3D WKT geometries. According to Mathias Bonduel[51] "The major benefit of this more lightweight approach for adding geometry is that any existing geometry format can be used directly. There is no need to develop a new geometry ontology and no additional toolchain for viewing and converting data is required."

The **Ontology for Managing Geometry (OMG)**[52] is a lightweight ontology focused on attaching geometry information to non-geometric objects, e.g., proving a building element with a mesh representation. The way this ontology is designed allows attaching geometric information in three different ways. The first and most straightforward way is to connect the object and its geometry directly. Alternatively, one can use an intermediary node between object and geometry to better handle things with multiple geometries. The final option is to use two additional nodes, one for handling multiple geometries and one to keep track of different versions of the geometry that are valid at different points in time. To sum up, OMG allows attaching geometric information to objects, handling multiple geometries for a single object and version history.

An ontology that is directly extending OMG is the **File Ontology for Geometry Formats (FOG)**[53]. Its only purpose is to specify the file formats in which a geometric representation is stored. This is particularly useful for users who try to get geometric information of an object in a file format they can open. Since there are many existing formats, most existing viewers only support a small portion of them. FOG is constantly extended through community efforts because new file formats and new versions of existing ones are constantly developed that need to be included in the ontology. Example relations defined in FOG are *as IFC v2x4*, *as STL format v1.0*, or *as Revit format*.

The closely related **Geometry Metadata Ontology (GOM)**[54] is directly compatible with both OMG and FOG. As the name already tells, GOM is concerned with defining concepts and relationships used for geometry metadata. This includes vocabulary to define length units, file sizes, and accuracy of geometric representations. Defining the coordinate systems and coordinate reference systems a geometry refers to is probably the essential part of the ontology. GOM allows specifying custom local coordinate systems, different types of references systems, and the transformations between them. Furthermore, GOM contains classes that give information about the relations between two types of geometric representations. For example that a given 2D representation of a building element is directly derived from a 3D Building Information Model.

## 5.1 Ontologies covering Site and static building models

**ISO 16739-1:2018 Open, international standard for data exchange:** Concepts as used in real-world building projects, for all building lifecycle phases.

In the last decades, the Building Information Modeling (BIM) concept has been advanced by the construction industry to catch up with the digitization movement[55]. In essence, BIM is about creating a database containing a digital building description including geometry and other semantic information such as component classification and properties. IFC is still one of the widely adopted data model for BIM approach. More recently, researchers have applied Semantic Web Technologies (SWT) and the concept of Linked Data, separately or in addition to conventional BIM systems, to address construction-related data integration, and interoperability issues[56]. **IFCOWL** is the semantic web counterpart to the EXPRESS schema definition of the Industry Foundation Classes (**IFC**) and formulated in Web Ontology Language (OWL). It is automatically generated from the EXPRESS file and has, except for some minor exceptions, the same content but is defined in a different language. IFC itself is an international open-source standard that specifies building and construction-related data. Its focus lies in the representation of geometric and semantic information of structural and spatial building elements but also includes, e.g., specifications regarding construction processes. Through means of IFCOWL, it is possible to make use of the vast number of classes defined in IFC and connect them with information from other domains using semantic web concepts[57]. Because of the large number of defined classes and the numerous connections between them, IFC (and consequently IFCOWL) is often considered to be too complex regarding most use cases needs.

The **Building Topology Ontology (BOT)**[58] is a minimal OWL DL ontology for defining relationships between the sub-components of a building. It was suggested as an extensible baseline for use along with more domain specific ontologies following general W3C principles of encouraging reuse and keeping the schema as simple as possible. Its primary application area is connecting sensor, product, and device data to building elements and spaces. Its most recent version includes as few as seven classes, fourteen properties, and one datatype property. Some of the most relevant classes like site, building, and storey are depicted in the graphic below. Apart from representing topological information, 3D information in the form of meshes can be attached[59]. Generally a BOT model is a simplified version of the IFCOWL model without going into detail about different subclasses of, e.g., building elements and almost completely neglecting geometric representations, which are a part of IFCOWL.

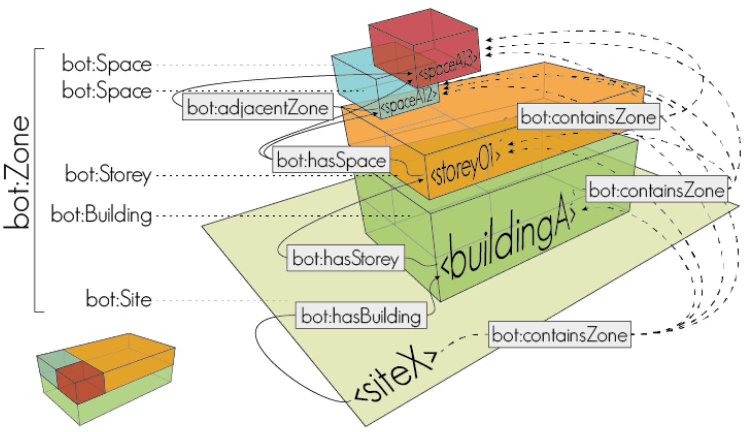


**Figure 15 - BOT Ontology**

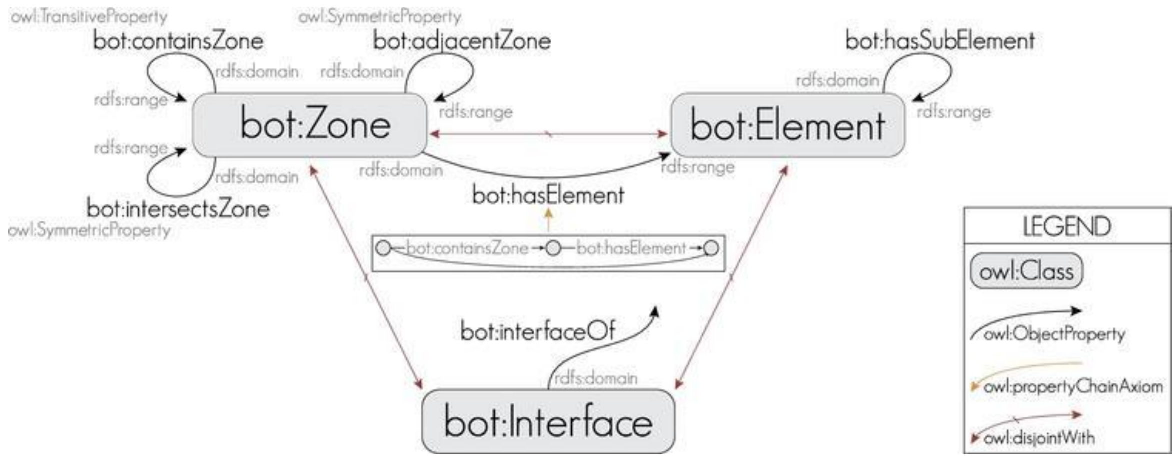


**Figure 16 - BOT Ontology (Focus on Element)**

BOT is interesting as a simplification of IFC where the level of information needed is low. For instance BOT does not describe material layers or boundaries.

**COBIEOWL**[60] is an ontology based on COBIe standard that defines non-geometrical properties of building products. Its primary purpose is transferring information from the planning and construction phase to the operational phase of a building. This can support facility managers and other stakeholders concerned with the operational phase with information about technical devices in buildings, warranty information, and maintenance-related data.

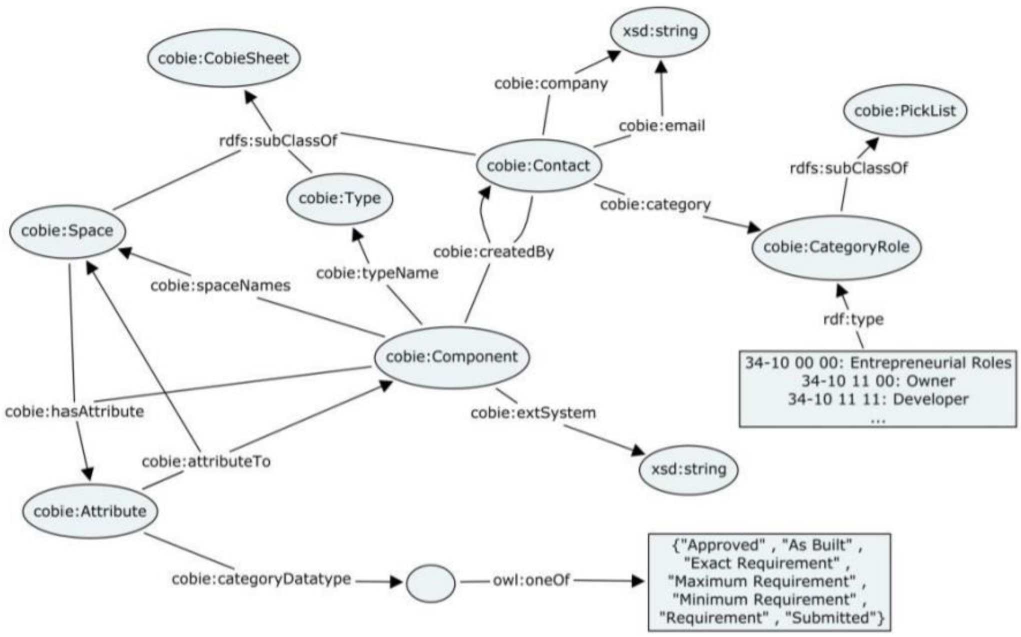


**Figure 17 - COBIE-OWL ontology**

COBIE is considered as a way to describe buildings and their components independently from the geometry. COBIE oriented toward data exchange rather than data modelling and he overlaps with IFC.

The two main relevant parts of IFC for BIM2TWIN model are the building decomposition with storeys, zones spaces, and building elements. COBIE is deconnected from building topology and IFCOWL is a complexe and monolithic ontology that imply a lot of inferences. For that reason, the use a more modular and lightweight ontology like BOT[61] with modular and linked ontologies -provided by Linked Building Data Community Group [62]- is prefered .

## 5.2 Ontologies covering Construction Processes and process monitoring

**ISO 29481-1:2010 "BIM – IDM – Part 1: Methodology and format".** Methodology to capture and specify processes and information exchanges during the lifecycle of a facility. ISO 29481-1:2010 defines unites the flow of construction processes with the specification of the information required by this flow, a form in which the information should be specified, and an appropriate way to map and describe the information processes within a construction life cycle.

**ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering workssets[63]** sets out the recommended concepts and principles for business processes across the built environment sector in support of the management and production of information during the life cycle of built assets (referred to as "information management") when using building information modelling (BIM). These processes can deliver beneficial business outcomes to asset owners/operators, clients, their supply chains and those involved in project funding including increase of opportunity, reduction of risk and reduction of cost through the production and use of asset and project information models.

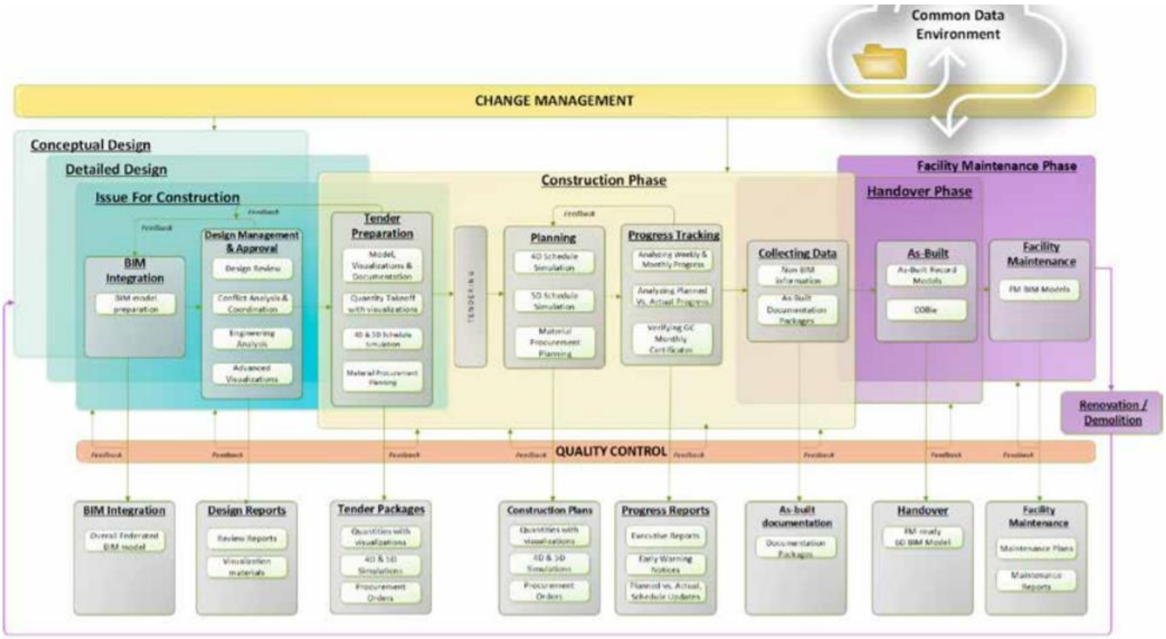


**Figure 18 - Flowchart illustrating lifecycle of a project and the coverage of a BIM execution plan**

Among documents required while implementing BIM in a construction projet, the BIM Execution Plan (BEP) is the most important. The BEP describes all aspects of BIM implementation on the project – from BIM uses and their integration into project management processes, over specific deliverables, analyses, interface management, communication and coordination plans facilitated by BIM, granulometry of the information to be delivered, to IT requirements, file formats and interoperability. BEP contains information on the project that should correspond to the Digital Twin implementation. Here are the main information that overlap with the scope of BIM2TWIN :

- project information
- project goals
- organisational roles/staffing

- BIM process design
- quality control
- model structure
- process tracking methods and reports plan
- data collection plan (for facility maintenance)

Roles and staff organization should correspond to **Agents** instances in the Digital Twin. **KPI**s in the Digital Twin have to be aligned with the projects goals. BIM process design, quality control and model structure have impact on the content of the BIM model that will instanciate the **As-designed Model** in the Digital Twin Platform. Process tracking and record plan should mention the use of a Digital Twin Platform with real-time data capture. At the end of the construction phase, the **As-built** part of the Digital Twin should be the mirror of the Building actually produced. This way the final **As-built Model** can be considered as the key common model with the Digital Building Twin in the maintenance phase.

**OWL-Time**[64] is an OWL-2 DL ontology of temporal concepts for describing the temporal properties of resources in the world or described in Web pages. It is a low level ontology useful for describing and keep track of dynamic events such as issues, on-site operations are mobile equipment location. The ontology provides a vocabulary for expressing facts about topological (ordering) relations among instants and intervals, together with information about durations, and about temporal position including date-time information. Time positions and durations may be expressed using either the conventional (Gregorian) calendar and clock or using another temporal reference system such as Unix-time, geologic time, or different calendars.

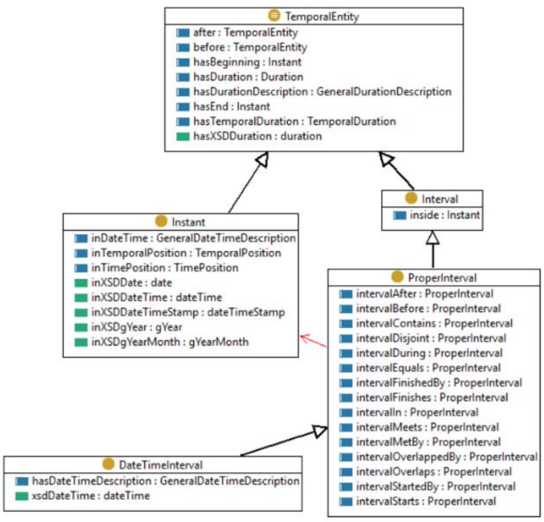


**Figure 19 - OWL-TIME ontology**

The **Semantic Sensor Network (SSN**) ontology is an ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators. SSN follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called **SOSA (Sensor, Observation, Sample, and Actuator)** for its elementary classes and properties.

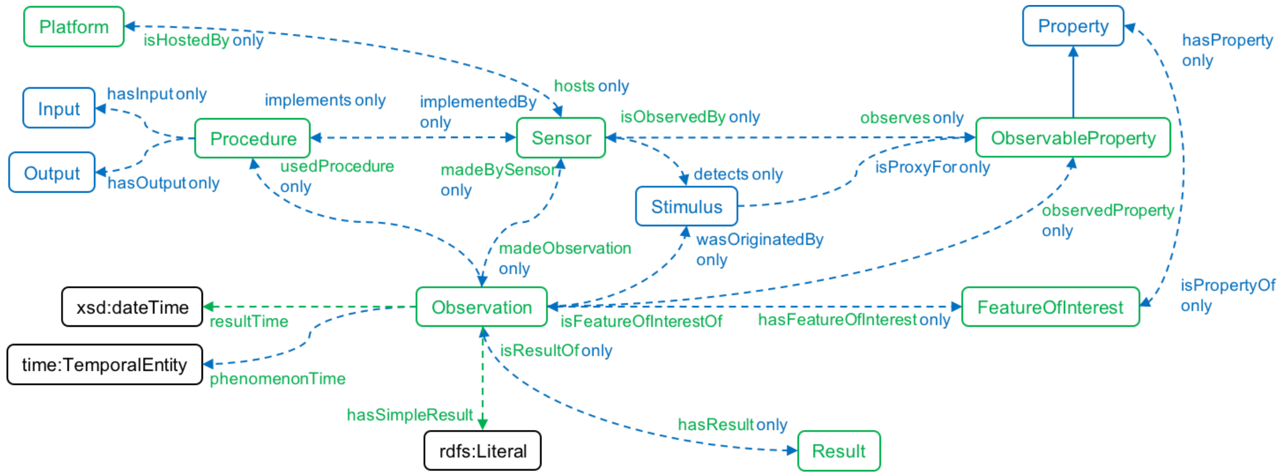


**Figure 20 - SOSA ontology**

SSN could be used to model the sensor network (camera, geolocation, lasers…) and smart devices (smart vehicles, connected tools) as a part of the construction site. SSN makes also links between sensors on their target by defining which properties are observed (for instance, the laser A measures the speed of vehicle B). SSN can provide a data framework for sensors installation and maintenance during the construction. It can also provide a way to keep track of the raw data sources.

Juan Du and Vijayan Sugumaran provides an ontology that covers **Supply Chain in Construction Domain** [65]. This ontology is dedicated to prefabricated components but some of the main concepts can be reused for raw material supply chain such as concrete. This paper underlines the complexity of supply chain and the heterogeneity of data, tools and agents that supply chain management involve.

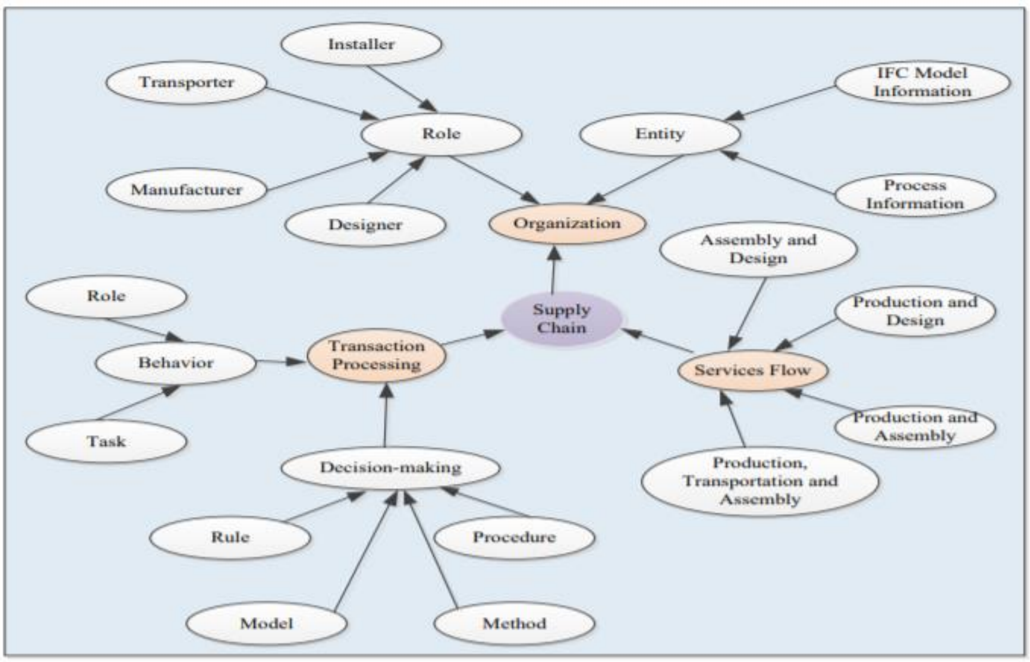


**Figure 21 - Supply Chain in Construction Domaine ontology**

In BIM2TWIN, only the construction site side of the supply chain will be modelled and limited to materials supply.

Amina Souag and Al provides a **Core Security Ontology**[66] that covers the security domain in its high-level aspects (threats and treatments) as well as its organizational ones (security procedures, security management process, assets, and persons). It can help in modelling sources of risks (for prevention), risk treatments (for improvements) as well as probable consequences of risks on persons or assets. This ontology is not dedicated to construction processes security, but the main concepts can be reused and extended for specific purposes.

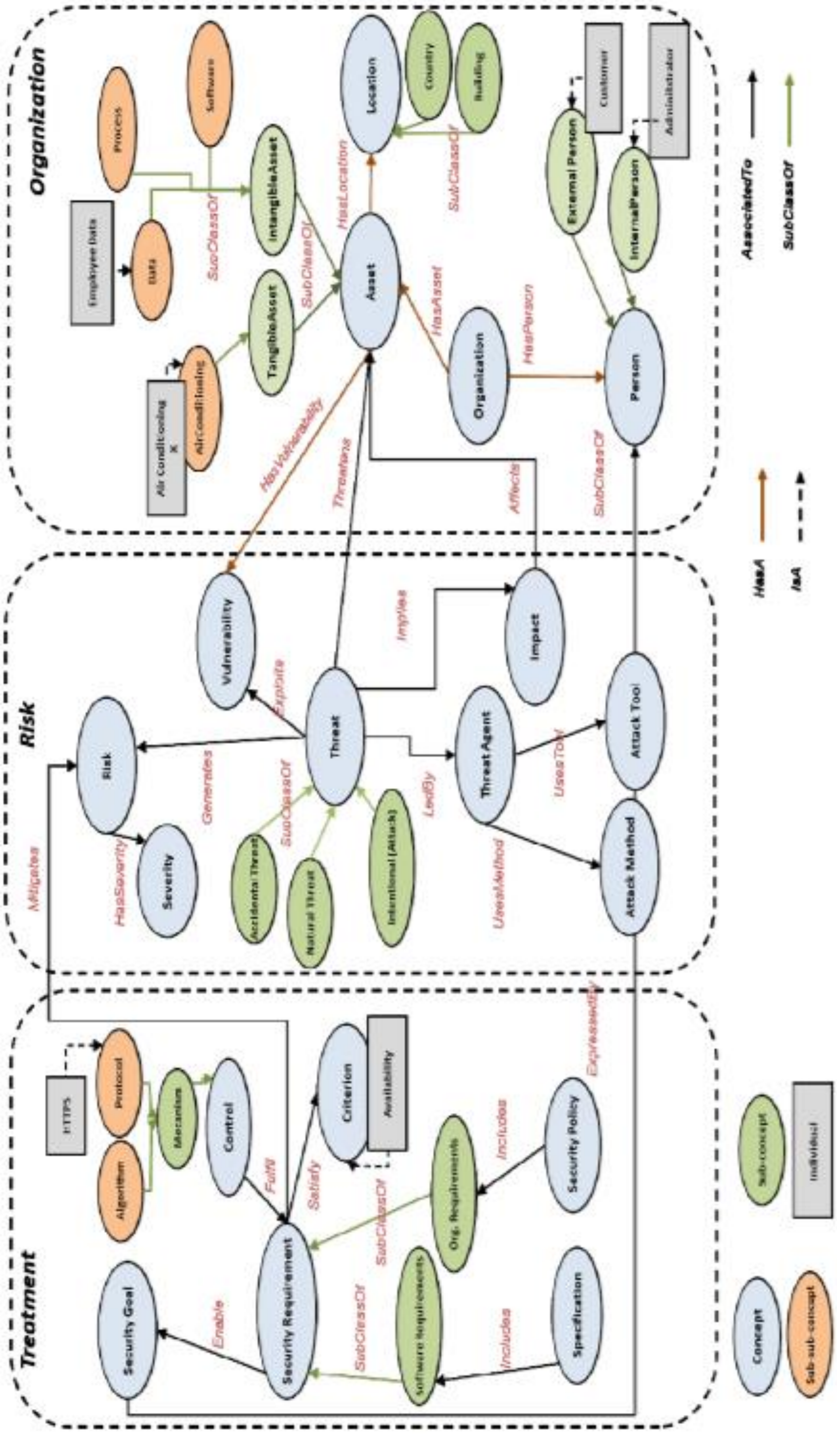


**Figure 22 - Core Security ontology**

The **Damage Topology Ontology (DOT)**[67] allows the definition of damage representations and their relations with other damages and affected construction components. The ontology supports a generic damage modelling approach and therefore could be applied for any type of degradation as well as for any construction type (e.g., buildings or bridges).

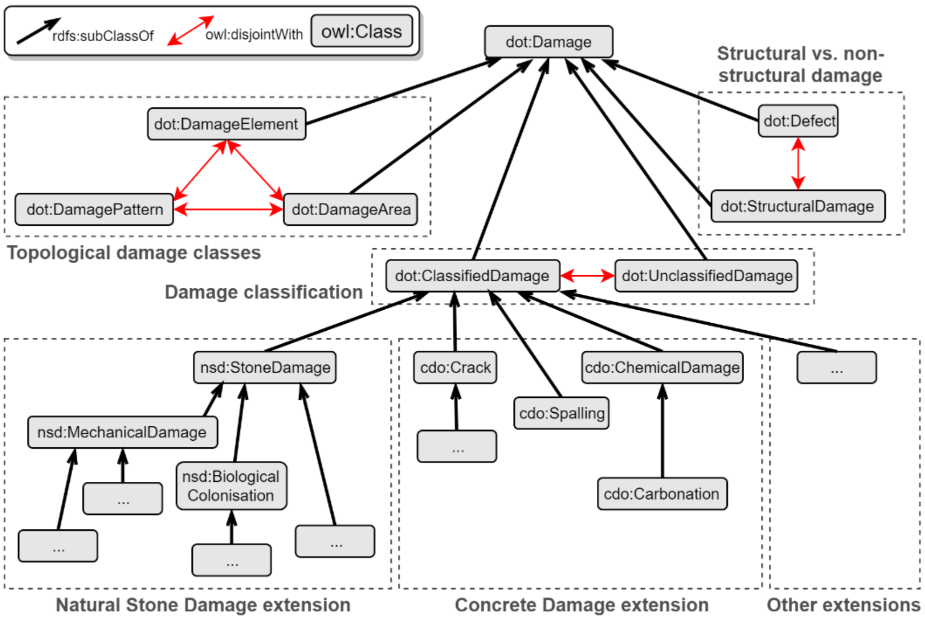


**Figure 23 - Damage Topology ontology**

Main principles of Damage Topology Ontology can be aligned to model quality defect in BIM2TWIN. But concepts might not be strongly aligned as damage in this ontology is considered as an unexpected change of state of an element due to an external factor whereas construction defect (BIM2TWIN concept) is an unexpected state during the production phase due to a failure in the construction process.

Du and Sugumaran provide an ontology that covers **supply chain in construction domain**[68]. This ontology is dedicated to prefabricated components but some of the main concepts can be reused for raw material supply chain such as concrete. This paper underlines the complexity of supply chain and the heterogeneity of data, tools and agents that supply chain management involve.

Business process (BP) modelling is an active area of research due to its multiple applications. For systems that support/monitor operators to perform their tasks (i.e., tasks of a given BP), a formal representation is essential. Various BP ontologies are available to formally represent BP. Various ontologies are covering BP domain, a first category of ontologies have been designed from scratch (PSL, COBRA, SMPO, IMAMO). A second category of ontology (BBO[69]) are various implementation of the **ISO/IEC 19510:2013**[70] standard and the associated **Business Process Model and Notation (BPMN)**[71]

**The Construction Task Ontology (CTO)** [72] describes tasks operating on construction elements, spatial zones and/or damages. The tasks are either planned or executed depending on the dataset metadata context of the dataset its used in. Five different types of tasks are defined: installation, removal, modification, reparation and inspection. Consequences of tasks on the dataset, i.e. added and/or deleted triples, are attribut using reified statements (RDF reification). The tasks can link to a reified statement using the CTO relations.

**Digital Construction Ontologies**[73] **(DiCon)** are shared representations of construction domain knowledge that specify the terms and relations of construction works and their related information. DiCon is based on a hybrid ontology development approach. It is composed of ontologies that aim to represent the objects, processes and actors that are relevant for the management and execution of construction projects. DiCon is aligned with lower-level ontologies such as FOAF (for agents), SOSA (for IoT), BOT (for Building Topology) and QUDT (for physical quantities). The DiCon includes six modules: Entities, Processes, Information, Agents, Materials, Variables, and Contexts. Concerning workflows, DiCon provides the concepts for the activities and their relationships between different entities (flows). The relationships are represented with variables and constraints. This ontology gives high level information concerning constraints, precondition, and dependencies of construction activity.

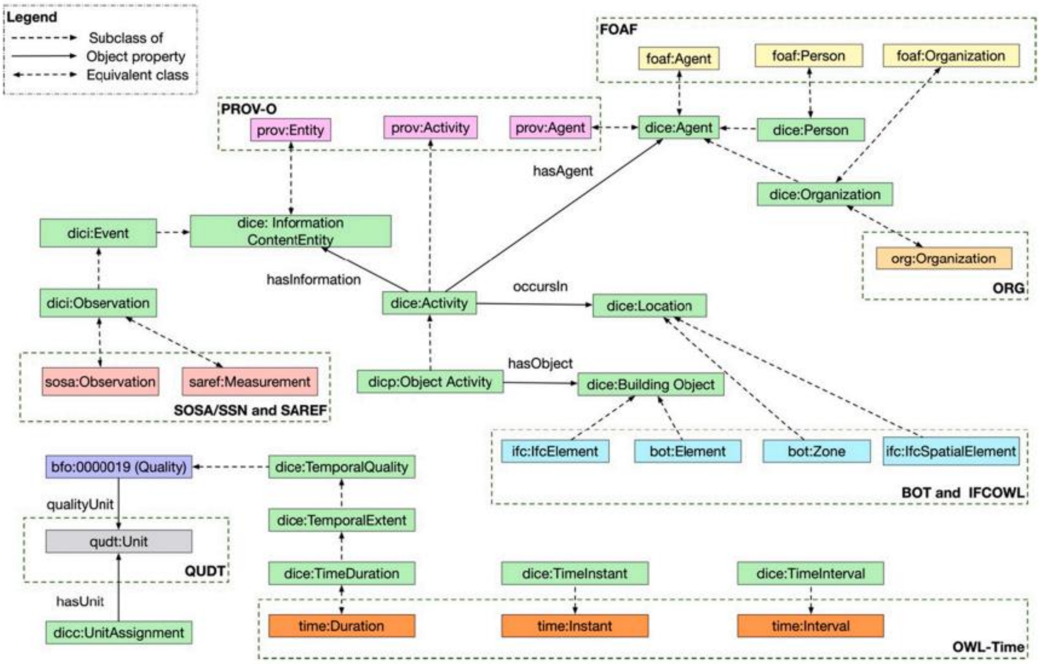


**Figure 24 - Digital Construction Process Ontology and relation between constraints, activities and conditions**

The Digital Construction Process Ontology provides the basic representation mechanisms for multi-context information that is commonplace in the construction domain: planned and actual values, as-designed and as-built models, and different levels of detail/development. The ontology allows the definition of different context frameworks, to create contexts, within the frameworks, associate content to contexts, and to compare objects and values across contexts. At the implementation level, the content is stored in the named graphs of an RDF Dataset. Since named graphs have identifiers, they can be associated with relevant metadata, such as provenance information or unit framework.

When compared to the main aspects investigated, it strikes that the aspect of Occupational Safety and Health of the personnel on site does not exist in DiCon.

## 5.3 Other lower-level ontologies

The following table shows general ontologies that are often reused and federated by industry oriented ontologies. Those ontologies provide basic vocabulary that are common with a wide range of industry domains and use cases.

**Table 4 - List of complementary and low level ontologies**

| Name | Description | Prefix | URI |
|---|---|---|---|
| Dublin Core | Metadata terms from Dublin Core | dct | http://purl.org/dc/terms/ |
| Schema.org | Search Engine metadata vocabulary | schema | https://schema.org/ |
| VANN | Annotations for examples and usgae notes | vann | http://purl.org/vocab/vann/ |
| VCARD | Digital standard for business cards | vcard | http://www.w3.org/2006/vcard/ns# |
| VOAF | Vocabulary description and their relations | voaf | http://purl.org/vocommons/voaf# |
| XSD | Extended Markup Language schema description | xsd | https://www.w3.org/1999/02/22-rdf-syntax-ns# |
| BFO | Fundamental categories (ISO/IEC 21838-2) | obo | http://purl.obolibrary.org/obo/ |
| OPM | Objectified property states with value calculation | opm | http://www.w3id.org/opm# |
| wgs84_pos | Basic geolocation for spatially-located things | geo | http://www.w3.org/2003/01/geo/wgs84_pos# |
| FOAF | Agents and social networks | foaf | http://xmlns.com/foaf/spec/ |
| Org | Organizations, posts, roles | org | https://www.w3.org/TR/vocab-org/ |
| DCAT | Interoperability between data catalogs | dcat | http://www.w3.org/ns/dcat# |
| OPM | Ontology for property management | opm | https://w3id.org/opm# |

## 5.4 Synthesis

This literature review shows that building scale modelling based on ontologies is well covered by recent research works. Modelling at larger scale also benefits from various National and European initiatives mainly on geospatial aspects.

**Table 5 - Analysis of ontologies scope and overlapping with BIM2TWIN scope**

| Ontology | Domain | Main concepts | Related B2T Concepts |
|---|---|---|---|
| INPIRES ontologies | Urban Digital Twin | Parcel, Building | Building, Site |
| IFCOWL | As-designed Building topology | Building, Space, BuildingElement | Building, BuildingElement |
| BOT | As-designed Building topology | Building, Space, BuildingElement | Building, BuildingElement |
| geoSPARQL | As-designed / as-built geometry | Point, Polygon | Geometry, bounding boxes |
| BEO | As-designed Building components | Product | Product, BuildingElement |
| Omniclass taxonomy | As-designed Building components | Product classification, Building Element classification | Product, Building Element |
| SSN | Sensor network | FeatureOfInterest, Observation, Sensor | Building Element, Resource, Equipement, Sensor |
| Core Security | Safety | Risk, Treatment | Person, Assests, Security Requirement |
| Damage Topology | As -built Quality | Defect | Defect, Building element |
| CTO | As-planned Processes | Activity, constraint | Activity precondition |
| Suppply Chain Ontology | Processes | Transaction, Role, Task, Process | Material, Role, Resource |
| OWL-time | Processes | Temporal Entity, Date | Acitivity, RawData |
| PROV ontology | As-performed Processes (Data), raw data layer | Entity, Activity, Agent | Agent, Activity, Resource, Eqiupement, Raw data source |
| SKOS | Taxonomies, alignments | Related concepts, Narrower concept | All concepts with taxonomies |
| QUDT | Quality, Material quantities, raw data layer, KPIs layer | Units, Physical quantities | All physical and theorical quantities with units |
| Dublin Core | Raw data layer | (Digital) resource | Raw dataset |
| KPI Ontology | KPIs layer | KPI, Goal, KPIValue | KPI |

Concerning the construction site scale, we noticed a lack of studies and researches. Concerning construction phases, conception and exploitation phases are more studied than construction phases that still need standards and specific model to be proposed. We also found that except for IOT domain, existing models are mainly static models that do not consider life cycle of resources. The opportunities to align BIM2TWIN domain with existing ontologies is discussed in part 8 dedicated to Logical Layer Integration.

## 6 ANALYSIS OF REQUIREMENTS: BRIDGING THE GAP BETWEEN THE IDEAL CONSTRUCTION PROCESSES AND ON-SITE REQUIREMENTS AND CONSTRAINTS

The main purpose of WP1 is to conceptualize the ideal Digital Twin construction process and define the main KPIs. This part makes a synthesis of this work by deriving the ideal model into a core model linked with on-site use cases such as quality monitoring, safety and optimization defined by WP3 to WP7. Whereas WP1 model view is driven by theorical approach, WP3 to WP7 took pilot constraints into account. The present approach is a trade-off between the ideal approach and a use case driven approach.

The following analysis is based on reports provided by the other work packages dedicated to specific use cases. The different work packages were still developing their use cases and data requirements during the development of the present document. For this reason we present only a first version of the integrated model of BIM2TWIN based on early analysis. We try to make the kernel of the model as stable as possible to preserve coherence when new classes and properties will be added.

### 6.1 Gathering vocabulary: the dictionary approach

The BIM2TWIN domain deals with a wide range of concepts. All those concepts can firstly be named without ambiguity and hierarchically organized; hence we developed a BIM2TWIN taxonomy based on the vocabulary used by practitioners and experts.


**Figure 25 - BIM2TWIN tag cloud**

We then organized the BIM2TWIN vocabulary in a dictionary by adding terms definition to the taxonomy. The BIM2TWIN dictionary[74] is built as a graph of terms and is based on the SKOS ontology. In this graph, terms are linked to each other according to the semantic relations defined by the SKOS ontology. The dictionary allows to share, update, browse and search the vocabulary for all BIM2TWIN partners. Each term of the dictionary comes with a definition in order to avoid any ambiguity. Terms are classified in four categories: **Classes** of objects, their **properties**, possible **relations** between classes and **taxonomies** that enrich classes hierarchy with abstract classes (with no specific properties). At this step of the BIM2TWIN project, the dictionary allows to precise vocabulary from raw terms to logical classes. An unambiguous and extendable classification system for construction components, task types, issues and resources should be defined. The taxonomies should be usable for classifying unique construction components and will be based on or interlinked with existing classification.

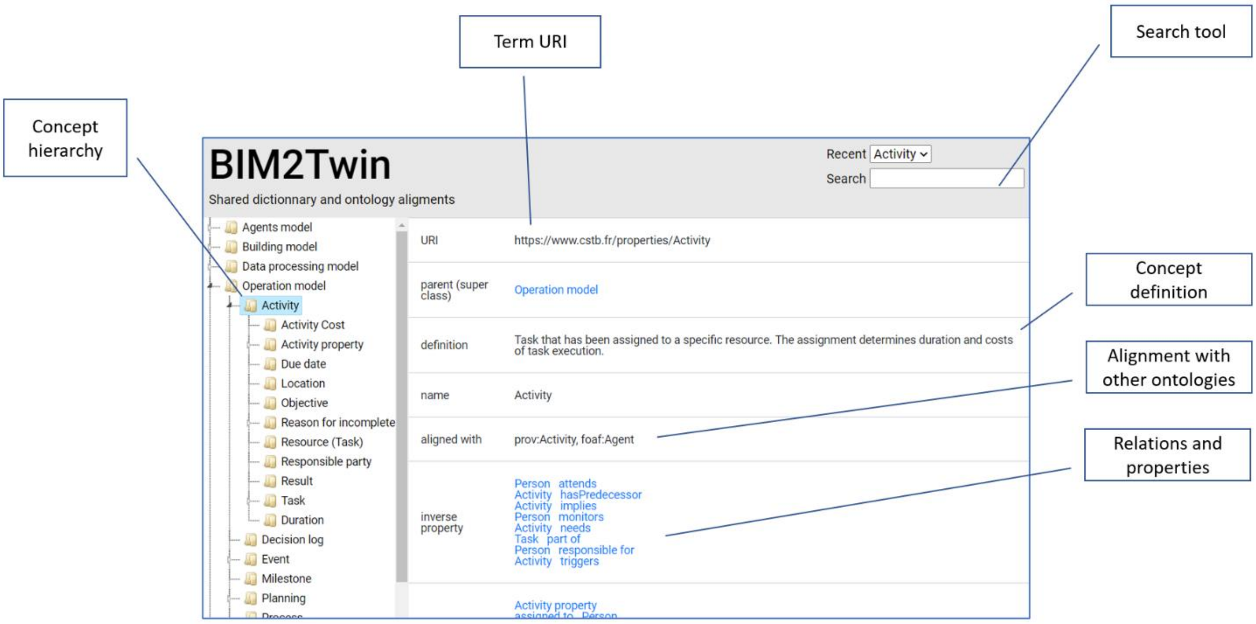


**Figure 26 - BIM2TWIN dictionary**

The dictionary is processable, it can be requested by other applications. We have built various indicators that monitor the dictionaries completeness and coherence. Those indicators are available through a web service updated in real-time.

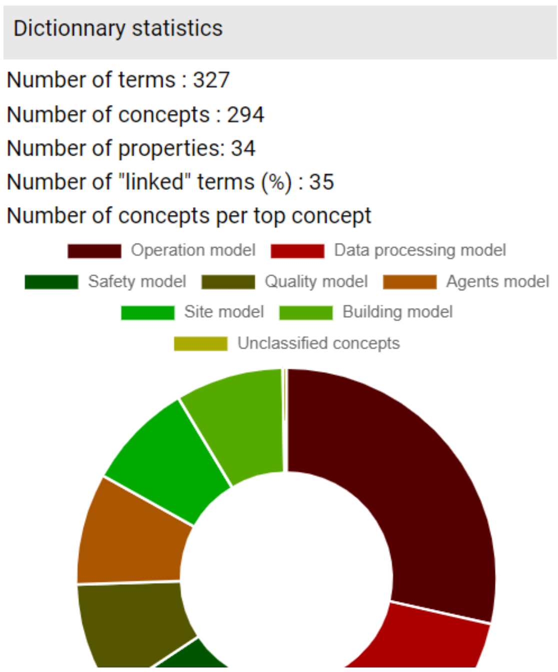


**Figure 27 - BIM2TWIN dictionary indicators**

The following parts introduce data models. Those data models use classes that are defined in the dictionary. Then the model will be enriched all along the project according to services needed by stakeholders. The dictionary will be kept synchronized with the model and will be used as a model documentation for the next steps of the BIM2TWIN project.

## 6.2 Core data model

A core data model was developed based on the UML models developed for the WP3 to WP7 with the use case driven approach, which are introduced in the subsequent sections. This model includes class definitions and their attributes that represent essential concepts relevant to all or at least multiple work packages. The core data model serves as the backbone of the digital twin in construction and can also be used for use cases that exceed the topics covered by WP3 to WP7. The UML models specific to progress and quality monitoring, safety, and optimization complement the core data model with classes that are particularly needed for the given use case. During the development of the model, special attention was put on making the processes on the construction site the primary elements of the model and appropriately include product information. To properly represent project intent information and project status information, the model was split into two parts: as-planned and as-performed. Both parts are explained in further detail in the following sections. Even though the two parts are displayed separately, there are clear connection points between the two. These are discussed later on. First testing of the model with real-world data has already been executed to determine if the developed concepts are appropriate for storing data from the construction site. Nevertheless, tests with more detailed information from past construction projects will be needed. Therefore, the model is subject to iterative improvement and will evolve over the entire BIM2TWIN project. The presented version only serves as a starting point. The goal of a construction process is to reach a functional and quality target while meeting cost and time constraints. The following figure gives an overview of the core data model with its two main parts. The core model is detailed in the next paragraph.

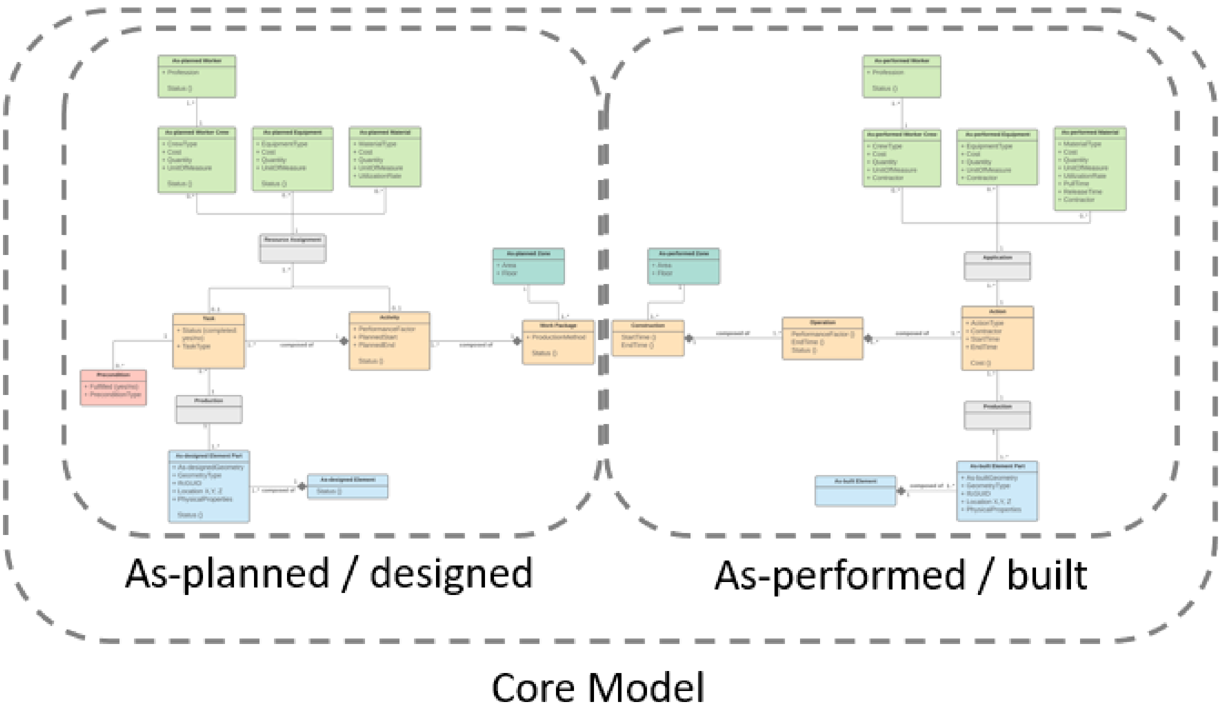


**Figure 28 - Core Model decomposition**

The digital twin logics need to embed the description of the target (**as-planned** processes and **as-designed** building) as well as the progression state of the process and built object (**as-performed** processes and **as-built** elements).

### 6.2.1 *As-planned / as-designed*

The main part of the as-planned section of the core data model is how processes are represented. They are divided into three levels. On the topmost level is the work package. It includes only very general information about the production method that is going to be used. For example, a work package instance in an imaginary construction project could be “construct all the columns on the first floor”. This would have additional information about the production method, e.g., the columns will be constructed through the cast-in-place procedure. One level below is the activity. Every work package is further divided into several activities that detail the planned start and end and the assumed performance factors. Staying with the example from before, an exemplary activity could be “place formwork for all the columns on the first floor”. There would also be an individual activity for the subsequent construction steps like casting concrete and removing formwork. The task represents the lowest level of the construction processes. It further separates the activities into tasks that refer to singular building elements so that every task is only connected to one as-designed element part. An exemplary task would be “place formwork for the column A1.1”. Since occasionally building elements modelled as one element are built as several elements, there is a possibility for aggregation with the classes as-designed element and as-designed element part. In some cases, activities are not further divided into tasks. This applies especially to preparatory work like excavations because they do not refer to a specific construction element.

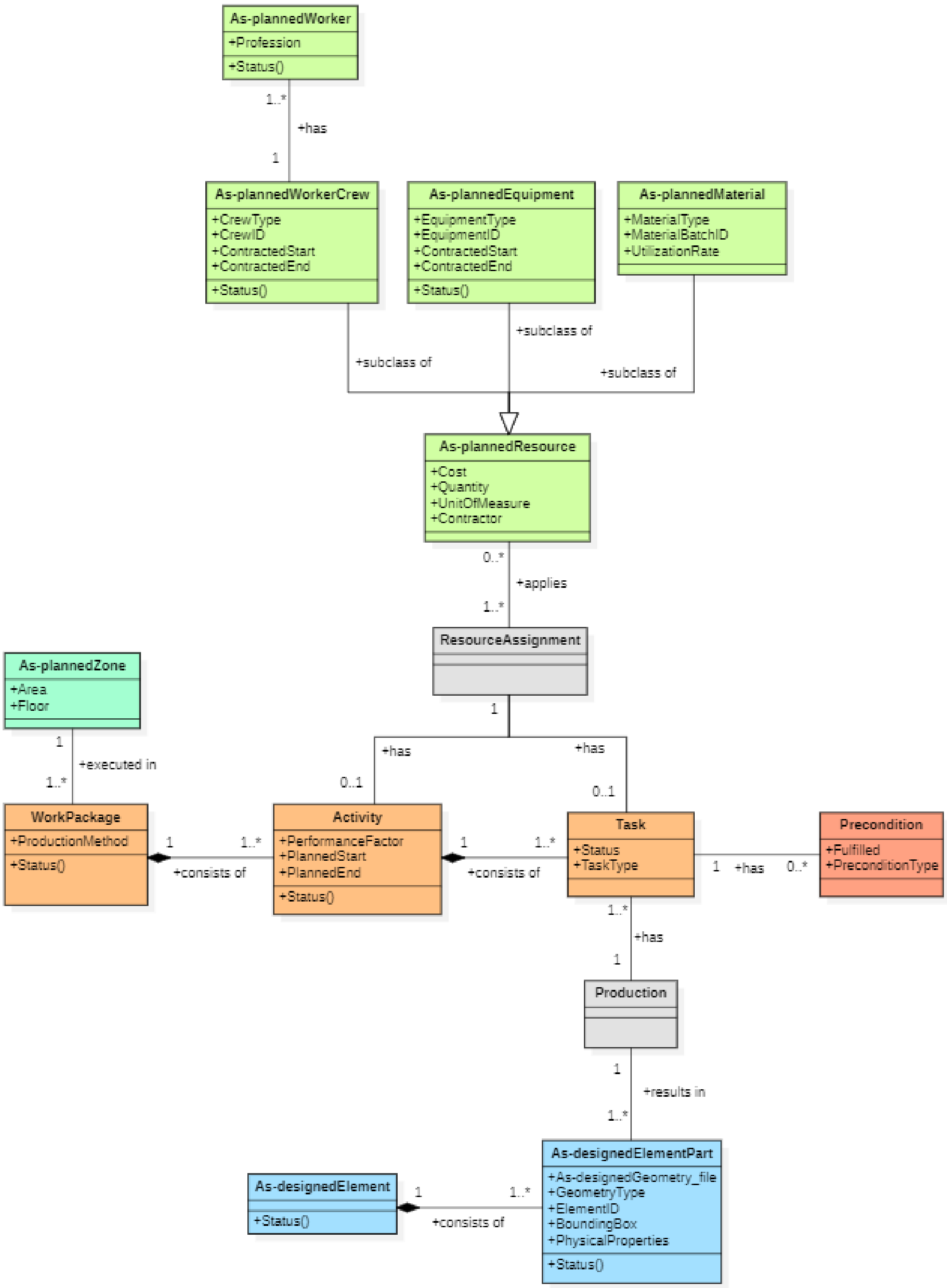


**Figure 29 - As planned / as-designed core model**

Tasks can also be connected with other tasks according to their precedence relationship, which defines the tasks that need to be finished as a requirement for another task to be able to start. As shown in the figure below, not only tasks can be a precondition for another task but also various types of resources and information that need to be available, environmental factors that need to be met, or zones that are required to be free to provide enough space for the working crew. The precondition class has a direct link to all these classes. For reasons of clarity, these connections are not directly shown in the as-planned UML diagram.

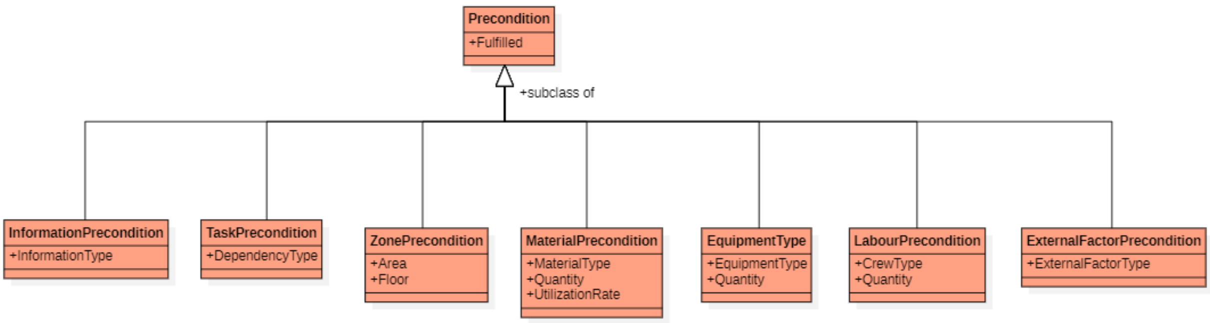


**Figure 30 - Precondition taxonomy**

Moreover, the task class is the origin of the status concept that originates from Huhnt[75]. Every task has the status “not started”, “in work”, or “finished”. As-designed elements and element parts also have a status that results from the task statuses of all the tasks the element relates to. Similarly, the status of an activity is also dependent on all the tasks it is related to. Only when all the tasks of an activity are finished the activity itself can also be considered to be completed.

Furthermore, resources like as-planned working crew, equipment, and material are part of the concept. On the one hand, the material directly connects to the task because it has access to the volumetric information of the building element that influences the amount of material needed. On the other hand, equipment and worker crew are connected to the activity one level above since they are usually not assigned to specific building elements but rather to a group of elements constructed as part of the same activity. Finally, the area a work package is dedicated to is defined by the as-planned zone.

It needs to be mentioned that even though all of the classes that were mentioned so far are part of the as-planned core model part, only a section includes the as-planned or as-designed prefix in their class name. All of the processes and preconditions have a different naming purposely or do not exist in the as-performed side of the model. For this reason, they can be told apart without a prefix.

### 6.2.2 *As-performed / as-built*

The *as-performed* model is structured very similarly to the *as-planned* model. Therefore, only the main differences between the two are explained. In terms of processes, the three levels also exist in the as-performed but are deliberately named slightly differently. Construction is the equivalent to the work package, operation is the equivalent to operation, and action is the equivalent to task. Instead of storing information about the planned construction process, they hold information about the actual start and end of the processes. Similarly, as-built elements store the geometry the way it exists on the construction site. In contrast to the as-planned model, the precondition class is omitted since this part of the model only focuses on the status of the construction site and not on the future plans. At last, significant differences to the as-planned model are the cardinalities of the process classes connected with other classes. Because of the nature of the construction site, where sometimes decisions are made very spontaneously, more complex combinations between processes and their building elements and resources are possible.

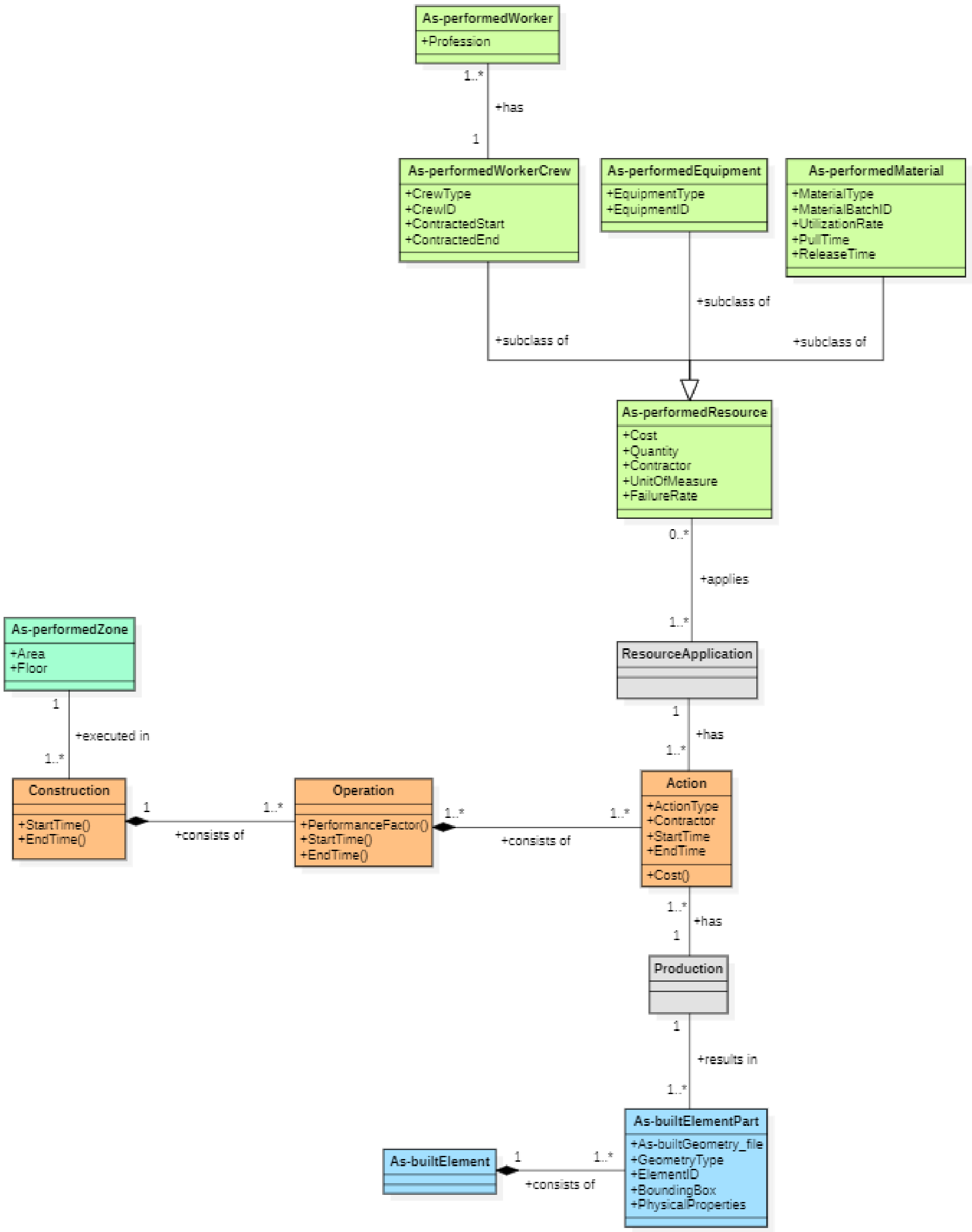


**Figure 31 - As performed / as-built core model**

### 6.2.3 *Connections between as-planned and as-performed model*

Although the two parts of the core data model are displayed separately, there are connections between them. One direct connection point are the classes construction and work package. It is expected that these barely differ in terms of the information they hold. Therefore, it could make sense to combine the two classes into one. For the purpose of visualization and differentiation between as-planned and as-performed, they are kept separately here. Other than that, every class from one side of the model is connected to its equivalent class from the other side, e.g., the as-planned equipment is directly related to the as-performed equipment. In an ideal world, both sides would be the same. Still, since the execution of construction projects is often far away from optimal, the two sides of the model allow to directly compare as-planned with as-performed and immediately figure out the deviations between them. We detail deviations in part 8 through the concept of Issue. At this point, it remains an open question if it makes sense to store information about deviations between the as-built and as-designed data and between as-performed and as-planned data in dedicated classes that would exist in an overlapping area between the two models sides. The alternative would be to compute the deviations on the fly by querying the digital twin graph database without storing them explicitly.

## 6.3 Quality monitoring (WP3 & 4)

Here-below are presented two use-cases related to deliverable "D3.1 – Requirements Specification for Digital Twin-Supported Progress Monitoring and Quality Control". The section focuses on how the data flow is handled from "raw data Layer" to "KPIs layer" with respect to the quality of objects placement and/or compliance of requirements between "as-designed" and "as-built" models. In each use-cases, classes, attributes, and relations between classes are presented.

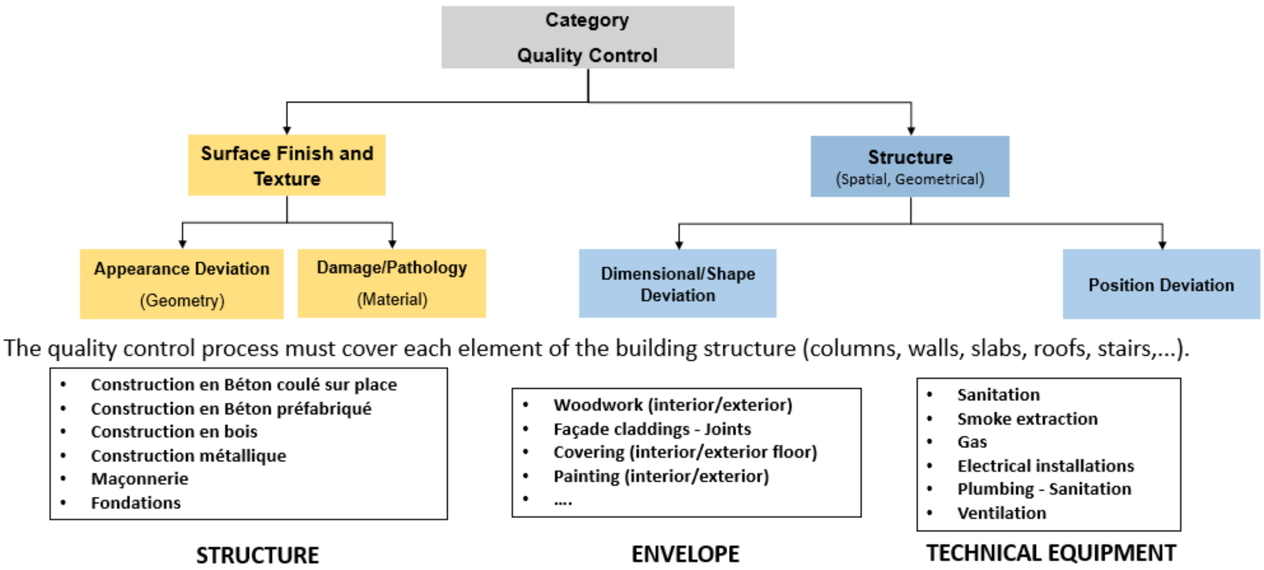


**Figure 32 - Quality controls and Elements taxonomies**

As often as possible, used classes in following UML diagrams are linked to dedicated ontologies, as mentioned in section §5 (ig. IfcOWL or BOT).

The here-after use cases are both fictional and take place on the construction site of an hotel composed of 5 storeys hosting each 16 chambers and 2 suites. The site is in the structural work phase: beams, walls and concrete slabs are under construction.

Then, a scan-vs-BIM approach is used : a dedicated algorithm recognises the ID of building element (from as planned BIM), its location, and checks the vertical alignment of points tagged as "walls". If between the lowest and the highest points of the wall, it exists a vertical deviation superior or equal to 0.001 m per m, a quality alert is sent.

*User Story 1: Concrete bugholes quality control*

According to the "as-planned" schedule, for the North aisle of the main building, the poured concrete walls must be exempted of surface bugholes (also called "surface voids") larger than 0.5cm of diameter. Quality monitoring is made by eye checking. Raw data are Boolean fields to be filled by an inspector in charge of the site quality. Data must be registered each day at 8:00 p.m. for each construction part ("as-built" Construction Element) built during the last 72h. Default value is authorized and is set with "False" (ig. No violation of the requirement).

If a defect is seen by the inspector, an image, used as visual proof, is registered in the data platform of the project.

**Figure 33 - Example of concrete bugholes**

The Table 6 shows classes, attributes and examples of values of the "Raw data layer" for the use-case #1. An inspector on the construction site fills a <u>check-list</u> dedicated to the *Daily visual Inspection of concrete <u>construction elements</u>.* The check-list is composed of <u>items</u> such as the Boolean question : Presence of concrete bugholes larger than 0.5cm of diameter. If the answer for a ConstructionElement if "True", then an <u>image</u> is taken (eg. Figure 33).

**Table 6 - Instantiation of the "Raw data layer" for use-case #1**

| Class | Attributes | Values |
|---|---|---|
| Checklist | ID | y5ezr54g6e4rg64 |
| | authorID | MathieuThorelCSTB |
| | name | Daily visual Inspection of concrete construction elements |
| | acquisitionTime | 2021-09-10 at 14:13:40 GTM |
| | comment | - |
| ChecklistItem | ID | 81gjom68Pd2Yd47 |
| | name | Presence of concrete bugholes larger than 0.5cm of diameter |
| | value | True |
| Image | ID | RC8h2s548t6kiAg |
| | authorID | MathieuThorelCSTB |
| | name | 4587921.png |
| | acquisitionTime | 2021-09-10 at 14:10:42 GTM |
| | comment | - |
| | format | PNG |
| | width | 1024px |
| | height | 720px |
| | ppp | 300 |
| ConstructionElement | ID | Uyg759f8c2sz67G |
| | ifcType | WallStandardCase |

The data passing throught the “raw datalayer” feed then the “Logical data layer” (Table 7). The captured bugholes on the ElementPart of the Wall are bigger than the geometricTolerance. Thus a SurfaceDefect is registered.

**Table 7 - Instantiation of the “Logical data layer” for use-case #1**

| Class | Attributes | Values |
|---|---|---|
| Wall | ID | Uyg759f8c2sz67G |
| | classification | wall |
| | reference | |
| | volume | |
| | cost | |
| | placement | |
| | dimension | |
| | orientation | |
| | isLoadBearing | True |
| | isExternal | True |
| | NetFootprintArea | 3.42 m |
| | NteSideArea | 58.4 m² |
| | NominalLength | 15.4 m |
| | NominalWidth | .07 m |
| ElementPart | ID | Uyg759f8c2sz615 |
| | classification | |
| | reference | |
| | creationTime | |
| | PercentageOfCompletion | |
| | rawDataLattestId | |
| GeometricTolerance | thresholdValue | 0.5 |
| | unit | m |
| Surface Defect | ID | 782thrd5487Etgk |
| | type | Concrete Hole on surface |
| | status | NOT_SOLVED |
| | date | 2021-09-10 at 14:10:42 GTM |
| | criticality | MEDIUM |
| | rawDataLatestId | 81gjom68Pd2Yd47 |

At last, on the “KPIs layer” of the workpackage 4 of the construction project, a NumberOfDefect if registered for a defined period of time (Table 8).

**Table 8 - Instantiation of the “KPIs layer” for use-case #1**

| Class | Attributes | Values |
|---|---|---|
| NumberOfDefect | ID | 058ynji59Dtig5I |
| | name | Number of defects |
| | startPeriod | 2021-01-01 |
| | endPeriod | 2021-12-31 |
| | value | 2 |
| | unit | - |
| | comment | - |
| | kpiFamily | Quality KPI |
| | workPackage | 4 |
| | criticalityLevel | MEDIUM |

*User Story 2: Vertical deviation of built walls*

According to the "as-planned" processes, for the North aisle of the main building, the poured concrete walls must have vertical deviation rate lower than 0.001 m per m. Quality monitoring is made by point clouds acquisition to check the potential vertical deviation of the wall. Raw data is Point Cloud acquisition. Datasets are updated each Monday afternoon with LiDar technology. In the Logical Layer, the Point Cloud scenes are analysed and each point is labelled with the "as-design" Construction Element ID. Quality control is done by the analysis of vertical alignment of labelled points cloud, by dedicated IA-based algorithms (not-defined in this document).

The Table 9 shows classes, attributes and examples of values of the "Raw data layer" for the use-case #2. A raw PointCloud acquisition of a ConstructionElement is made during the *Daily visual Inspection of concrete construction elements* thank to LidarScanner.

**Table 9 - Instantiation of the "Raw data layer" for use-case #2**

| Class | Attributes | Values |
|---|---|---|
| PointCloud | ID | a5ezr54yhe4rg67 |
| | authorID | MathieuThorelCSTB |
| | name | Daily visual Inspection of concrete construction elements |
| | acquisitionTime | 2021-09-10 at 14:13:40 GTM |
| | comment | - |
| | numberPoints | 5.789.000 |
| | boundingBox3D | MultiPolygon(x1 y1 z1, x2 y2 z3) |
| | colorSystem | RGB |
| | intensity | None |
| | type | Colorized Lidar Points Cloud |
| LidarScanner | ID | u1gjom58472Yd47 |
| | type | Time-of-Flight |
| | precision | 3mm |
| | distanceUnit | meters |
| ConstructionElement | ID | Uyg759f8c2sz67G |
| | ifcType | WallStandardCase |

The data passing throught the "raw datalayer" feed then the "Logical data layer" (Table 10). The captured geometry of ElementPart of the Wall shows a bigger vertical deviation than the geometricTolerance. Thus a VolumeDefect is registered.

**Table 10 - Instantiation of the "Logical data layer" for use-case #2**

| Class | Attributes | Values |
|---|---|---|
| Wall | ID | Uyg759f8c2sz67G |
| | classification | wall |
| | reference | |
| | volume | |
| | cost | |
| | placement | |
| | dimension | |
| | orientation | |
| | isLoadBearing | True |

| | | |
|---|---|---|
| | isExternal | True |
| | NetFootprintArea | 3.42 m |
| | NteSideArea | 58.4 m² |
| | NominalLength | 15.4 m |
| | NominalWidth | .07 m |
| ElementPart | ID | Uyg759f8c2sz615 |
| | classification | |
| | reference | |
| | creationTime | |
| | PercentageOfCompletion | |
| | rawDataLattestId | |
| GeometricTolerance | thresholdValue | 0.001 |
| | unit | m per m |
| Volume Defect | ID | 782thrd5487Etgk |
| | type | Vertical deviation |
| | status | NOT_SOLVED |
| | date | 2021-09-10 at 14:10:42 GTM |
| | criticality | HIGH |

At last, on the “KPIs layer” of the workpackage 4 of the construction project, a NumberOfDefect if registered for a defined period of time (Table 11).

**Table 11 - Instantiation of the “Raw data layer” for use-case #2**

| Class | Attributes | Values |
|---|---|---|
| NumberOfDefect | ID | 9uiOpji59Dtig5I |
| | name | Number of defects |
| | startPeriod | 2021-01-01 |
| | endPeriod | 2021-12-31 |
| | value | 0 |
| | unit | - |
| | comment | - |
| | kpiFamily | Quality KPI |
| | workPackage | 4 |
| | criticalityLevel | HIGH |

The three following UML diagrams refeer to the class layers (raw data layer, logical layer, KPIs Quality Layer) of the User Stories #1 and #2, previously described.

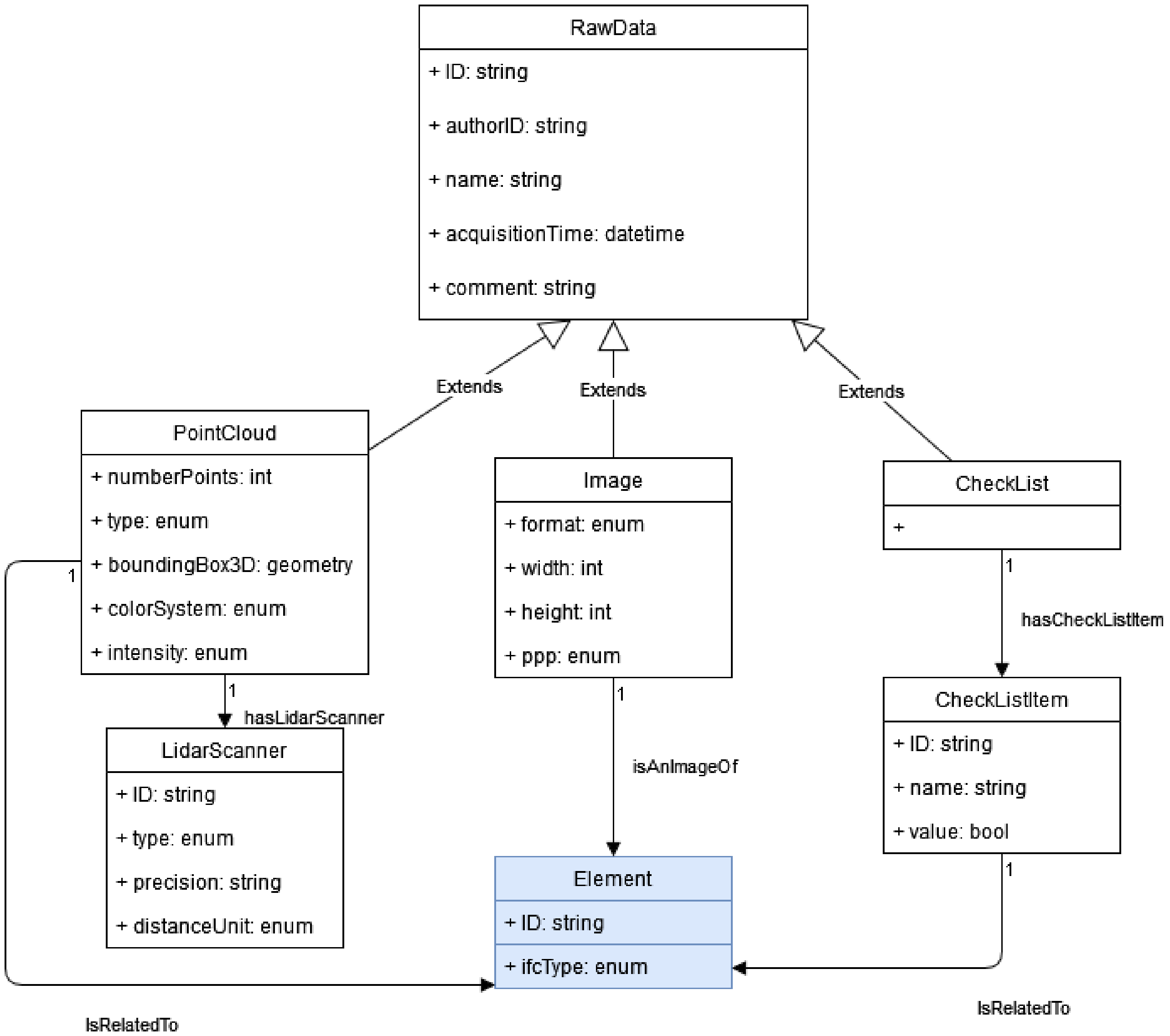


**Figure 34 - "Raw data layer" for use-cases**

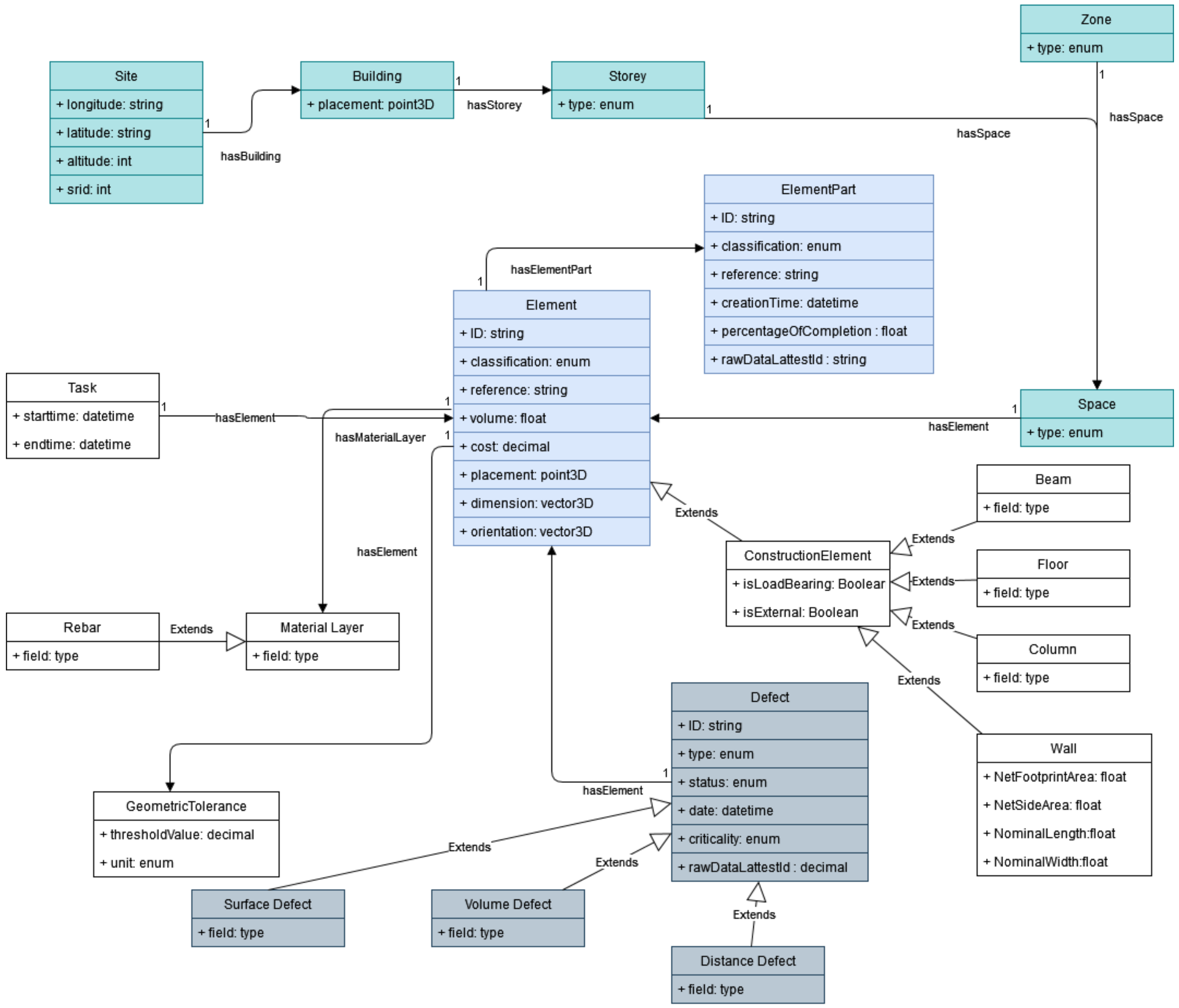


**Figure 35 - “Logical data layer” for use-cases**

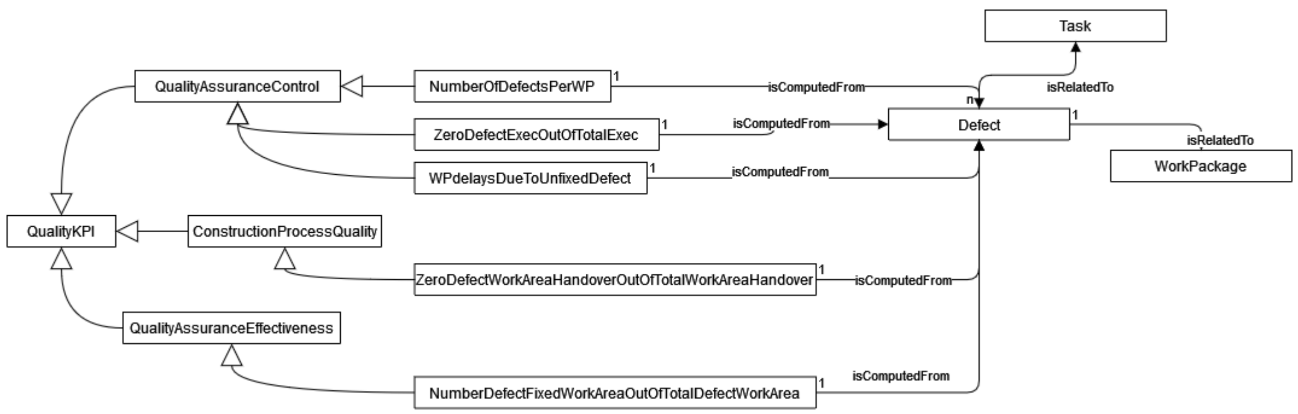


**Figure 36 - “KPIs Quality layer” for use-cases**

## 6.4 Safety (WP5)

WP5 is focussed on the safety of construction workers. It consists of 3 main individual components that enhance and evaluate safety of construction plans, detect conformance of the construction site with planned safety measures, classifies and severity of discrepancies and intervenes to mitigate and analyse incidents. The "Prevention through design and planning" (PtD/P) module adds safety measures to alternate plans which are generated in WP7, evaluates their progress and their overall safety to facilitate the plan selection of the decision maker. The "Conformance checking" module observes the implementation of safety measures in the construction process with the help of WP3 and WP4 and classifies the severity of discrepancies. The "Right time analysis and mitigation" module provides alarms in a suitable format and timely manner to different actors in the construction process to prevent incidents and allow for investigation of accidents. An additional module integrates VR training of workers taking into account the current state of the construction site and upcoming planned tasks.

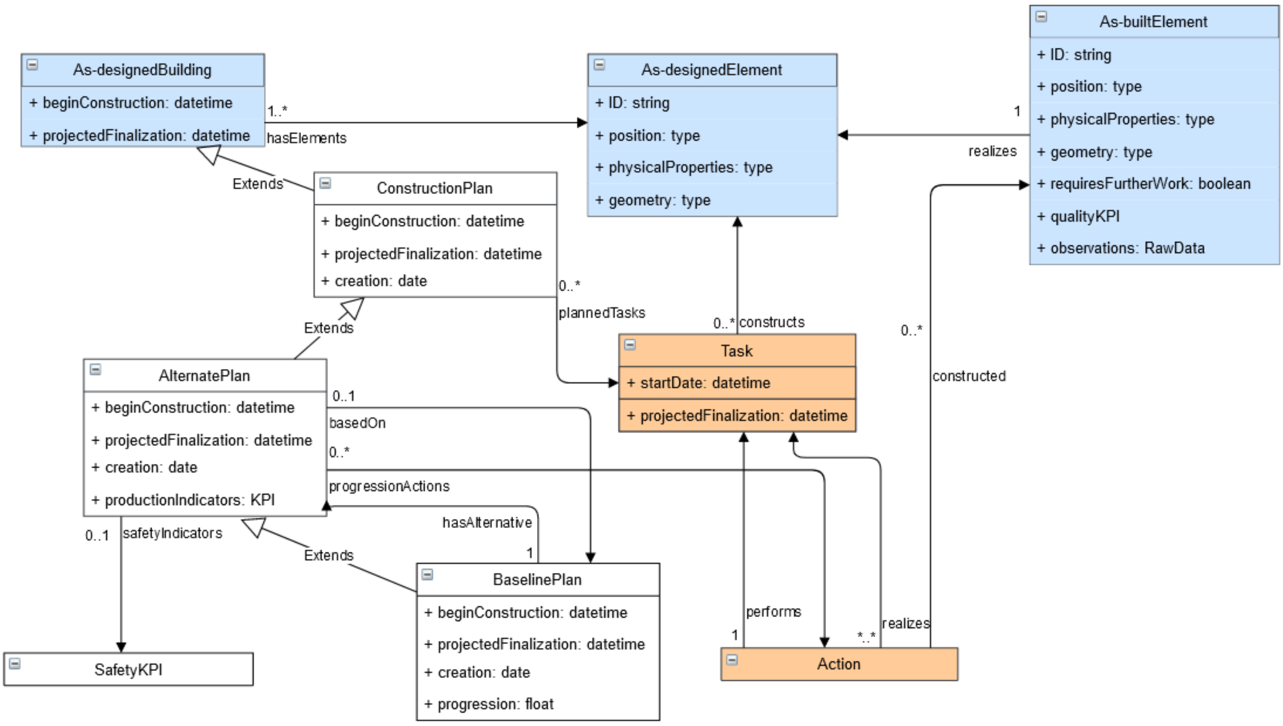


**Figure 37: Relation between Alternate- and BaselinePlan**

The AlternatePlan and the BaselinePlan are the central components in the PtD/P module. AlternatePlans are generated throughout the construction process and thus contain Tasks as well as Actions. They are generated in WP7 based on the current BaselinePlan and enriched with safety measures in the PtD/P module. The PtD/P also identifies the HazardZones caused by each Task. The estimation of KPIs and SafetyKPIs based on safety rules and best practices helps the decision maker to chose the next BaselinePlan.

The "Conformance checking" module accesses the BaselinePlan and RawData to utilize methods developed in WP3 and WP4 to verify the employment of safety equipment and safety installations. SafetyDiscrepancies are generated based on detected deviations and classified using SafetyRules and BestPractices.

The "Right time analysis and mitigation" module takes into account the BaselinePlan, SafetyDiscrepancies, processed and unprocessed raw data collected from the construction site in order to alert workers on the construction site for immediate prevention and to inform the decision makers. The history of events and SafetyDiscrepancies allows an investigation of the root cause of accidents and contributes to the prevention of future incidents.

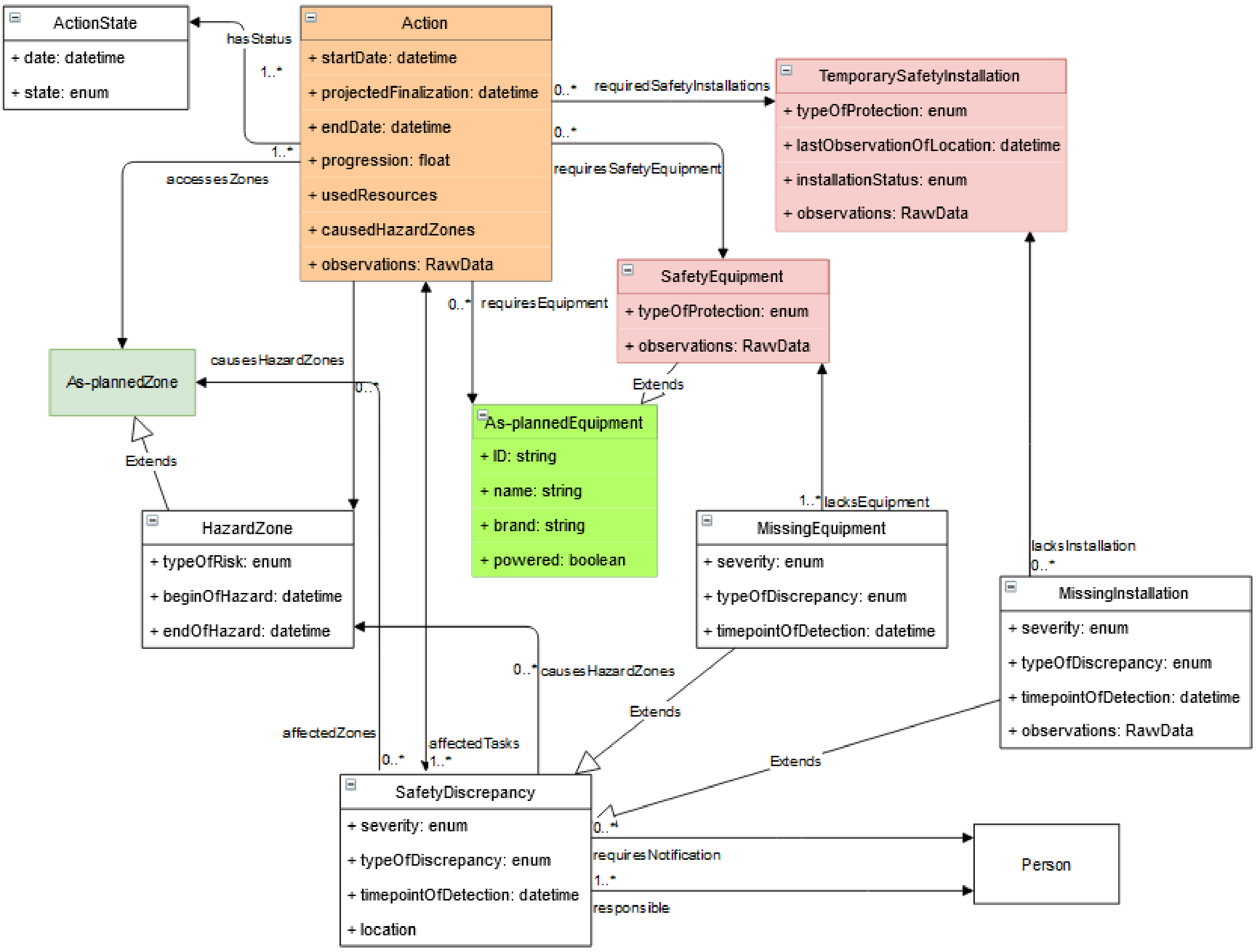


**Figure 38: Actions, their required SafetyEquipment/-Installations and interaction with Zones/Hazardzones.**

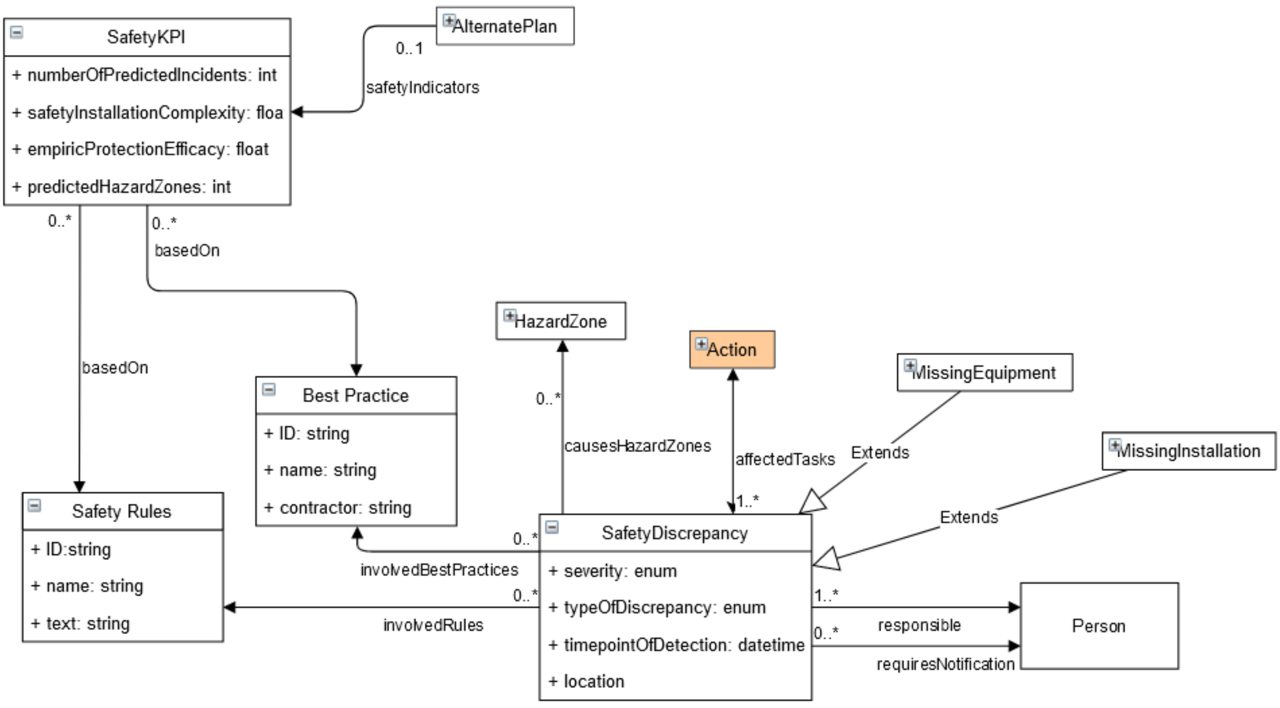


**Figure 39: SafetyDiscrepancies often involve the violation of SafetyRules and BestPractices. They may help to improve latter and thus help a better estimation of SafetyKPIs.**

*User Story 1: Guardrail and formwork compliance checking*

Formwork and guardrails are essential installations on construction sites that not just provide access to certain places, but also give safety for the workers themselves and below. Putting them into place correctly, however, is necessary and not always easy due to various different elements in different sizes. The verification of installed objects is done on raw data collected onsite, i.e., point cloud data and images.

Assuming a safe BaselinePlan generated by WP7 and the PtD/P module as well as a recently acquired point cloud, the "Comformance checking" module will use the RawData and the methods of WP3 and WP4 to confirm the presence of TemporarySafetyInstallations, i.e., guardrails and formwork in this case. The timestamp of the lastObservationOfLocation and the installationStatus within the TemporarySafetyInstallation is updated. In case of absence a MissingInstallation class, a specialisation of a SafetyDiscrepancy, is created. The severity and typeOfDiscrepancy are determined by the "Conformance checking" module. The detection of safety discrepancies triggers the "Right time analysis and mitigation" module, which sends short and quickly comprehensible notifications to affected workers from affectedTasks and affectedZones.

*User Story 2: Ground vehicle moving towards pedestrian worker*

The construction of a building involves a larger number of workers and different equipment and large amounts of material. Vehicules are usually involved in moving equipment, material and sometimes also workers around the construction site. As the sight may be limited due to clutter on the site, accidents may happen. A larger number of actors, i.e., workers and vehicles, requires real-time tracking and thus excludes point clouds. Often cameras can be attached to the building or mounted on a crane to gain overview over the site.

Similar to User Story 1, the arrival of RawData triggers the processing to check the conformance of the site as well as to detect objects, in this case AI-based object detection for vehicules, persons and moving objects. The detection of a car moving through unexpected zones or moving quickly triggers the instantiation of a SafetyDiscrepancy object noting the location. If the corresponding task involving a car can be associated an immediate notification is sent to the operator of the vehicule to reduce its speed. The SafetyDiscrepancy creates HazardZones and Zones overlapping with other tasks can be found. Thus, a notification to warn affected involved workers can be send to lower the risk of an accident. A direct identification of a worker via AI seems unreliable and thus a direct notification of a worker is difficult. The SafetyDiscrepancy, the involved risk and the effect on tasks can be used to adapt the BestPractices.

## 6.5 Equipment optimization (WP6)

WP6 concerns optimizing the use of equipment on construction sites. Four use cases are derived based on the types of interactions between the equipment and the DBT. Each use case requires a unique set of information from the DBT to perform optimization locally. The actions and status of the equipment are captured through equipment sensors and delivered to the DBT as structured information. Such information is expected to be translated into project status information and knowledge for analysis.

The first use case relates to the **positioning** of equipment on the construction site. Here, the positions of the equipment should be determined in real-time with the estimation of possible movements. Once the position of a piece of equipment has been captured, the location is to be displayed on a digital site plan (see Figure 40). To achieve this, 3D positions are calculated by using multiple construction site cameras.

The captured positions will be attached with time stamps. The type of equipment and a unique ID are also stored in a database and shared with other processes. Both the type and the ID are to be transmitted by the equipment, starting from the moment it enters the site, via wireless communication. If this is not possible, computer vision methods can be used to determine the type. To obtain an ID, a QR code must alternatively be visibly attached to the equipment. This code will be read out by a camera.

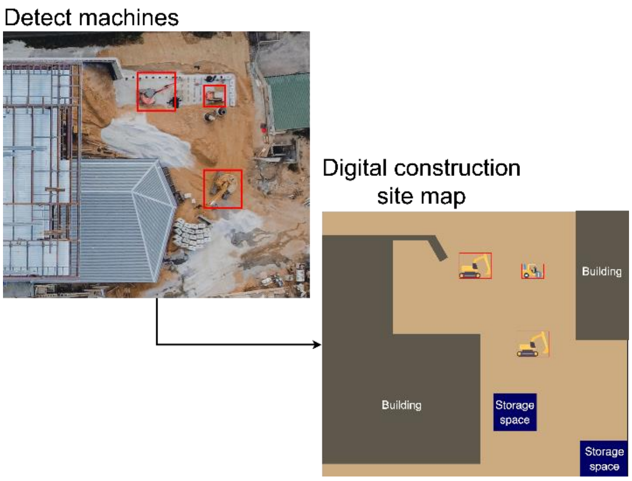


**Figure 40. Detected equipment is displayed on a digital construction site map**

The second use case refers to the **status** of the equipment. Here, a distinction is to be made between the four following states:

- Idle
- Work
- Travel
- Malfunction

With the registration of the status, it is possible to carry out a target/actual comparison. For example, a comparison can be carried out between the work record of a piece of equipment and its job schedule to measure discrepancies between the utilization rate and actions performed. The schedule could be updated based on the comparison if the status was not planned in this way.

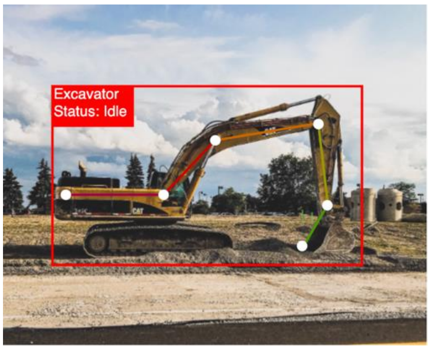


**Figure 41. Exemplary status detection with a computer vision method**

The third use case relates to the **movement** of equipment. It should be possible to determine if the path is blocked by something. For this, the equipment's surroundings must be monitored. Here, it is planned to attach cameras to the equipment. Using computer vision methods, obstacles or hazards should be detected, as shown in Figure 42, and reported to the DBT platform. To be able to quickly detect obstacles, the detection should be done locally (equipment is equipped with a computer) at best. This allows the equipment to send the message directly to the DBT platform.

By reporting detected problems to the site management, solutions can be found faster. This mechanism enables dynamic response to unforeseen problems [70]. For example, if an equipment has detected a blocked path on a large construction site, an intelligent equipment could reschedule the transport route. The new proposed route is shown to other machine operators on a display, and they can follow it.


**Figure 42. Detection of hazards**

The last general use case is about **communication** between the equipment and the DBT platform (see Figure 43). The central component of this use case is the communication with the DBT platform. It is planned that task data and schedules are exchanged between the equipment and the DBT platform. At this point, it is assumed that a human reads the job data and performs the work. Also, the detected problems from the movement use case will be communicated through the same communication interface. In addition to communicating with the DBT platform, smart equipment should also communicate with each other. The idea is to communicate data through a common interface so that equipment can coordinate with each other. If it is planned that two machines work together, the platform will communicate the ID of the partner machine beforehand.

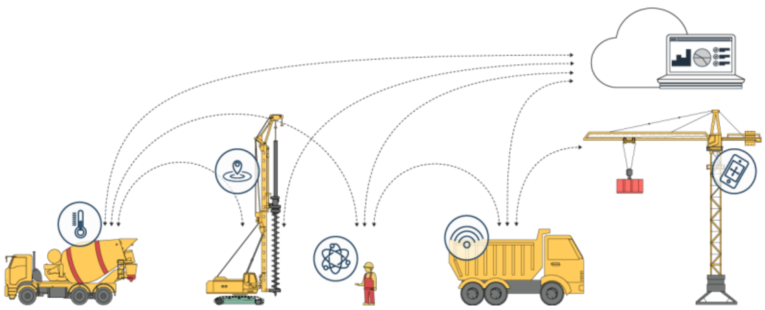

**Figure 43. Communication between the equipment and the platform on the construction site**

Based on the above four use cases, the required data schema of DBT for equipment optimization is derived. In the conceptual domain of project intent information (PII) (Figure 44), there is the *As-designed Site Layout* composing of the *Location Breakdown Structure*, *As-designed Travel Route*, *As-designed Work Location*, and *As-designed Hazardous Zones*. The latter three intersects with the data class of *Zone,* which composes the *Location Breakdown Structure* and can be contained by itself. The organization of *Zones* defines the spatial organization of the construction site for planning and control. Meanwhile, the DBT also contains the *As-planned Equipment Job*, representing a scheduled executable task for a specific piece of equipment. This job operates on *As-planned Material* and through *As-designed Travel Route*. Moreover, this job composes the *As-planned Equipment Job List*, which is sent to the designated *As-performed Equipment* (Figure 45).

Parallel to the as-designed counterpart, the *As-performed Site Layout* in the project status information (PSI) is composed of *As-performed Travel Route, Work Location, and Hazardous Zone*, all of which intersects with the *Zones* in the *Location Breakdown Structure* in the PII. In addition to these three, the *As-performed Site Layout* also comprises the *As-built Static Structure*, which contains the geometry of all static elements on the construction site, including *As-built Element Parts*, *As-performed Material*, and *As-performed Static Equipment*. Lastly, the *As-performed Site Layout* is translated into a *Digital Construction Site Map* to be sent to the *As-performed Equipment* along with the job list from the PII.

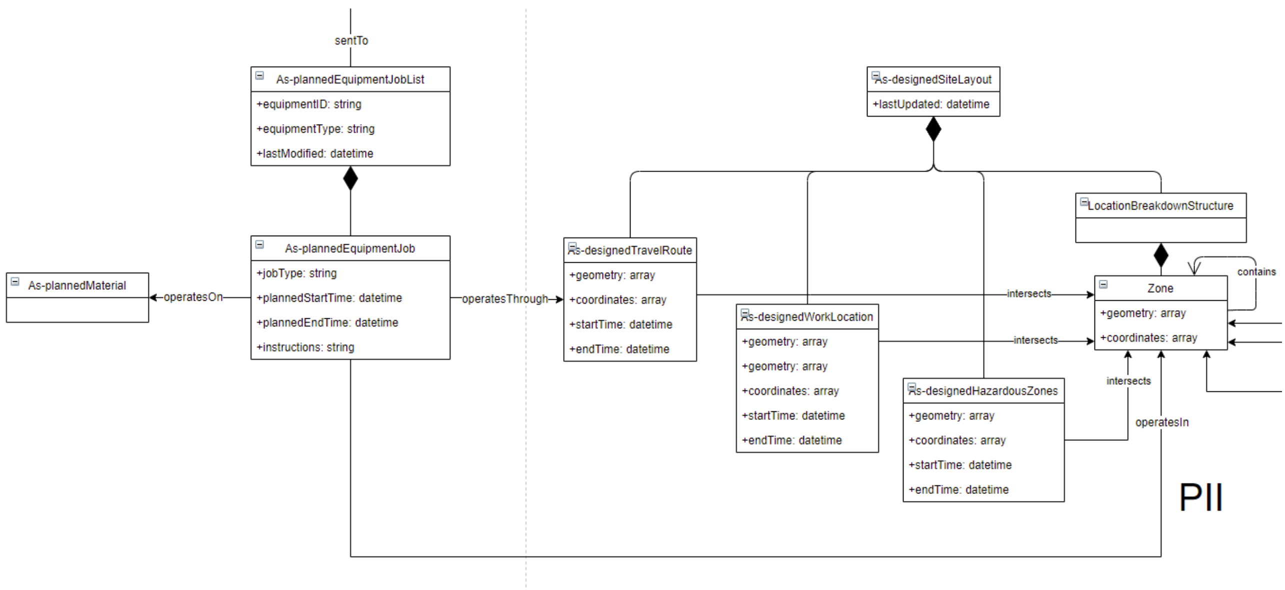


**Figure 44. Required Project Intent Information Schema for WP6**

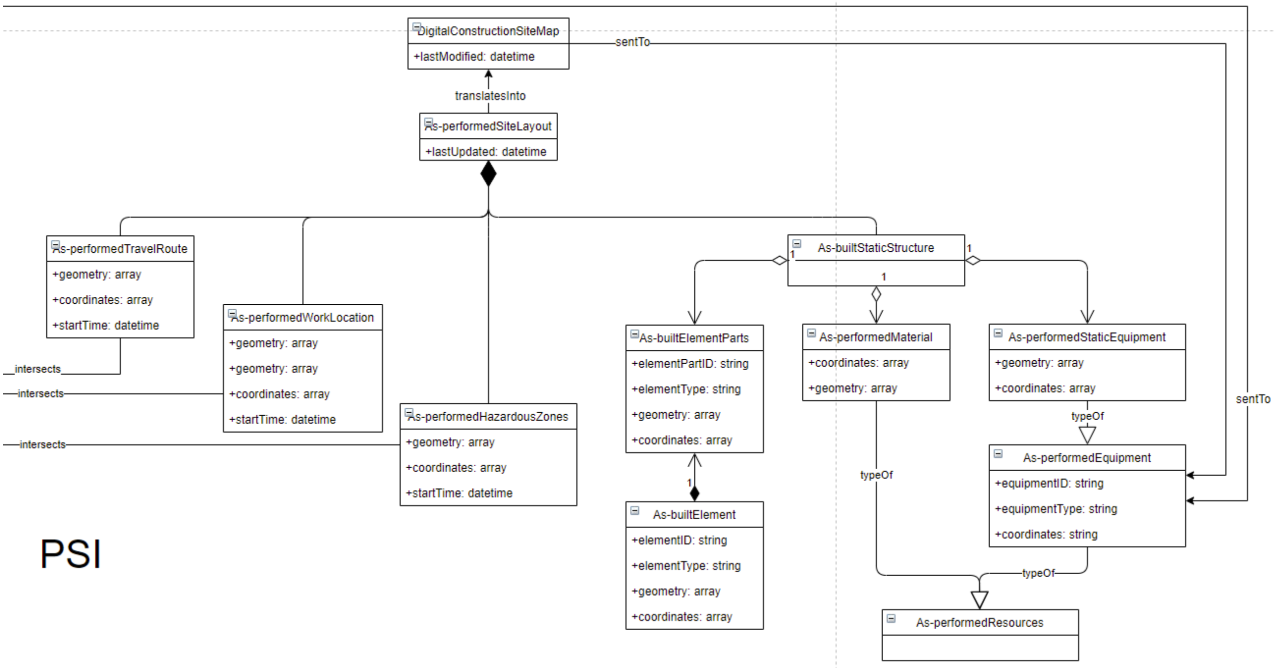


**Figure 45. Required Project Status Information Schema for WP6**

WP6 utilizes four types of equipment sensors to track the operations and status of the equipment:

- Motion sensor
- Weight sensor
- GPS sensor
- Pose estimation sensor

These sensors deliver raw data (Figure 46) to the DBT in the form of *Images, Load data, Position data,* and *RFID logs*.

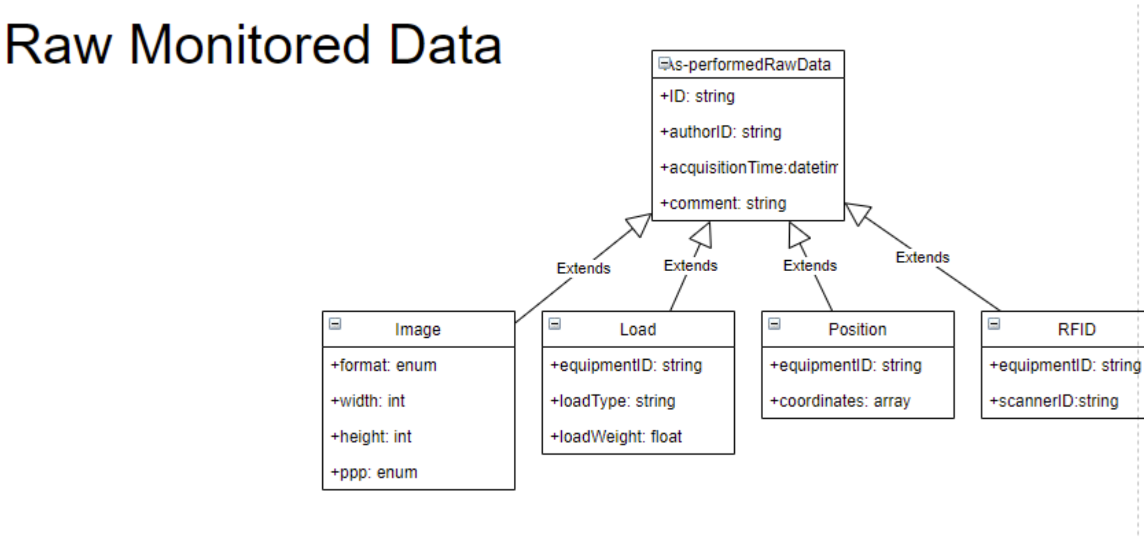


**Figure 46. Required Raw monitored data schema for WP6**

## 6.6 Process optimization (WP7)

WP7 is focusing on process optimization with the following goals :

- Waste reduction
- Meeting deadlines
- Optimizing cashflow
- Resource optimization (maximizing utilization percentage)

By taking into account planned tasks and resources availability (workers, materials, equipment).

Here are some user stories that illustrates those goals and constraints

The as-planned process model (**Figure 47**) contains the construction plan's work breakdown structure (WBS) and its associated information. These data represent the as-planned processes in a construction project and belong to the PII. Going down the level of granularity, the WBS comprises Work Package, Activity, and Task, with each composed of instances of the latter. A Work Package is an executable construction deliverable that describes in detail a specific scope of work, and is usually restricted to the work of a single skilled crew in a single project location. Activities detail the construction method of the Work Package they compose – they are the operations, while the Tasks outline the specific deliverable objectives in an Activity. A Task applies the work of an operation to a specific building element, or to part of a building element.

For example, the Work Package of "exterior structural walls in zone B of floor 2" is composed of construction activities "first-side formwork," "rebar fixing," "close formwork", "pour concrete" and "strip formwork." Each activity applies to the full set of building elements included in the Work Package. Within the activity of "first-side formwork" are the construction tasks associated with the individual building elements: "first-side formwork for Column 1," "formwork for Column 2," etc.

A Work Package is typically associated with user-defined milestones such as handover dates. Work Packages should be associated with As-planned Zones, including As-planned Work Location, As-planned Hazardous Zone, and As-planned Travel Route. An As-planned Work Location defines where the associated work should be performed and is an essential component for Location-based Scheduling [19]. An As-planned Hazardous Zone is an area determined as unsafe, liable to safety hazards. Such a zone changes dynamically through time according to the construction activities in and around it. An As-planned Travel Route can refer to the operation route of equipment, such as the crane radius, or the travel route of work crews and materials.

Constraints, or preconditions, play an essential role in construction projects' performance since they highly affect their cost and duration. To have efficient constraint management, implementation of constraint-based planning is critical to enhance planning reliability, reduce uncertainties in construction processes, and increase the transparency of project management. More specifically, construction tasks require all constraints to be fulfilled and eliminated before they are begun. According to lean construction principles, insufficient planning at the task level is often the single most important factor that affects project workflow breakdowns.

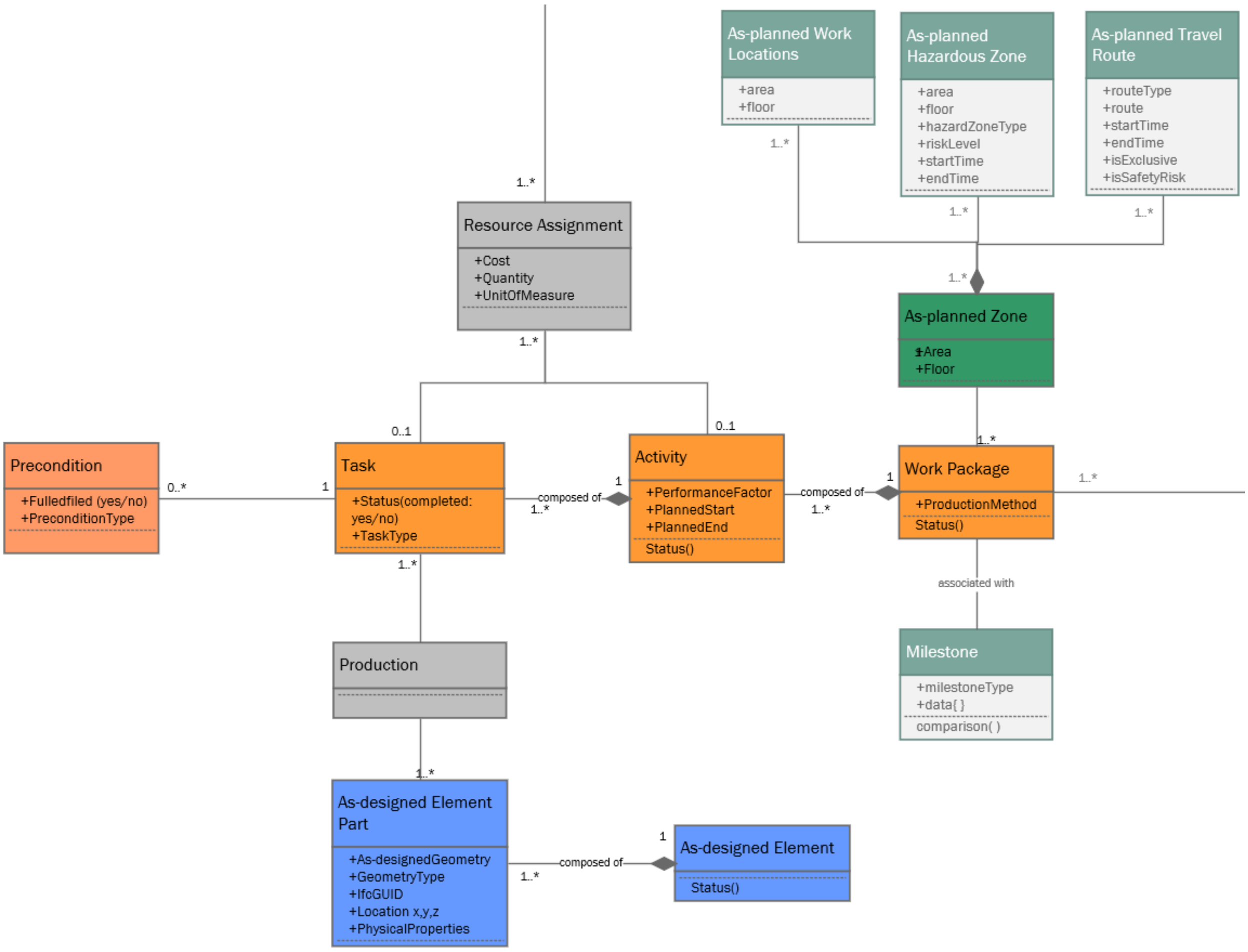


**Figure 47. UML class diagram for as-planned process model**

Figure 48 presents the data schema for as-planned resource assignment, which captures the intended resource allocation through time in a construction project. As-planned Resources are contextualized within the construction schedule through associations with Tasks and Activities (Figure 47) via the relational class Resource Assignment. These associations are inherited by the sub-classes of As-planned Work Crew, As-planned Equipment, and As-planned Material Batch, which represents the resource types of labor, equipment, and material, respectively. Additionally, As-planned Equipment is associated with an As-planned Equipment Maintenance Schedule, while a material batch is associated with a delivery schedule.

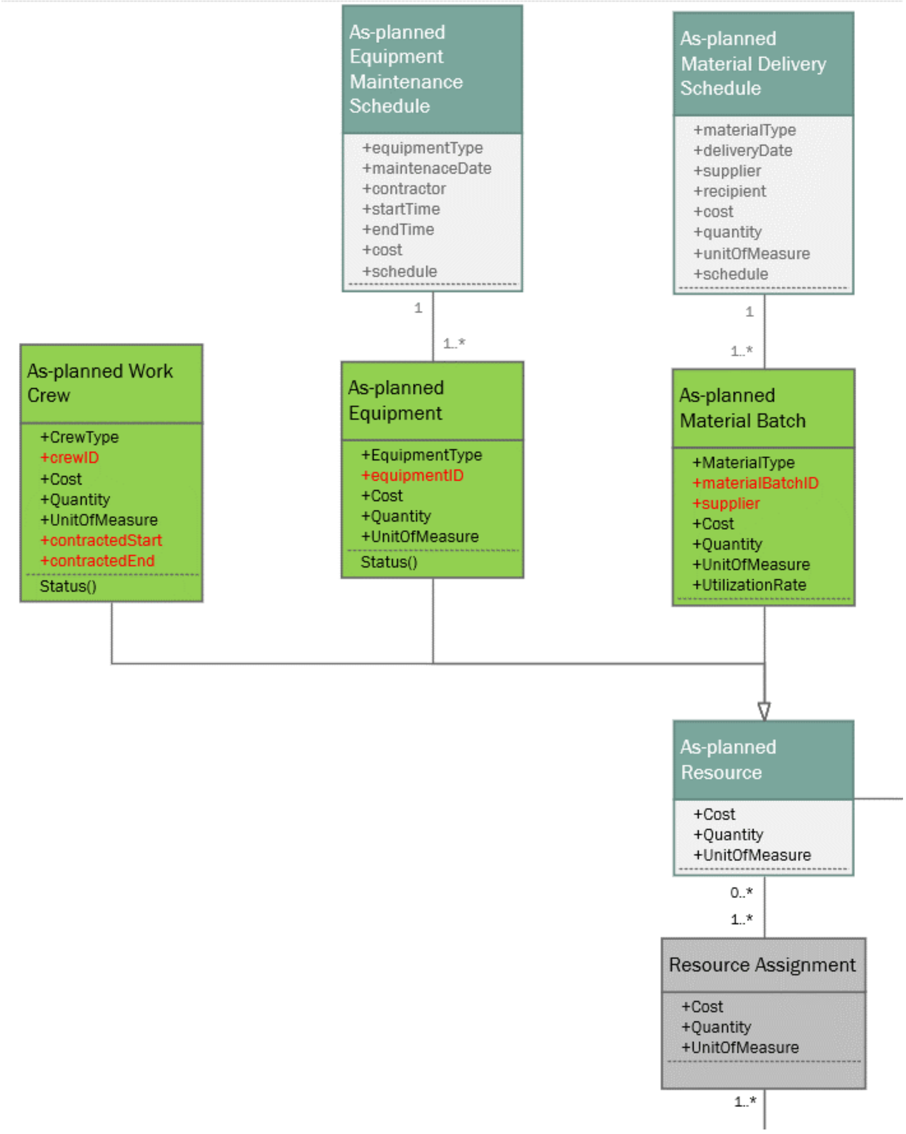


**Figure 48. UML class diagram for as-planned resource assignment**

As provided by the WP7 software, simulation input and output (Figure 49) contribute to the knowledge in the execution performance and thus constitute part of the PIK within the DBTP. A simulation-based planning cycle is represented through the Simulation Cycle class, composed of Simulation Parameters, Simulation Raw Output, Evaluation Result, and Decision Aids. Since these components are only of interest to the WP7 planning software, we envision that their data can be stored in the form of linked files.

Meanwhile, a Simulation Cycle is also associated with As-planned Construction Plans (BP or Aps) composed of Work Packages and their subsequent associations. "Construction plan" is a term intentionally chosen to highlight the structural similarity between BP and AP. Apart from the domain they belong to – BP in PII and AP in PIK – BP and AP are indistinguishable in their data structure, and it is this characteristic that enables the seamless updating of the current BP with the selected AP in a planning cycle, as discussed in Section 4.3.2 and 7.5.

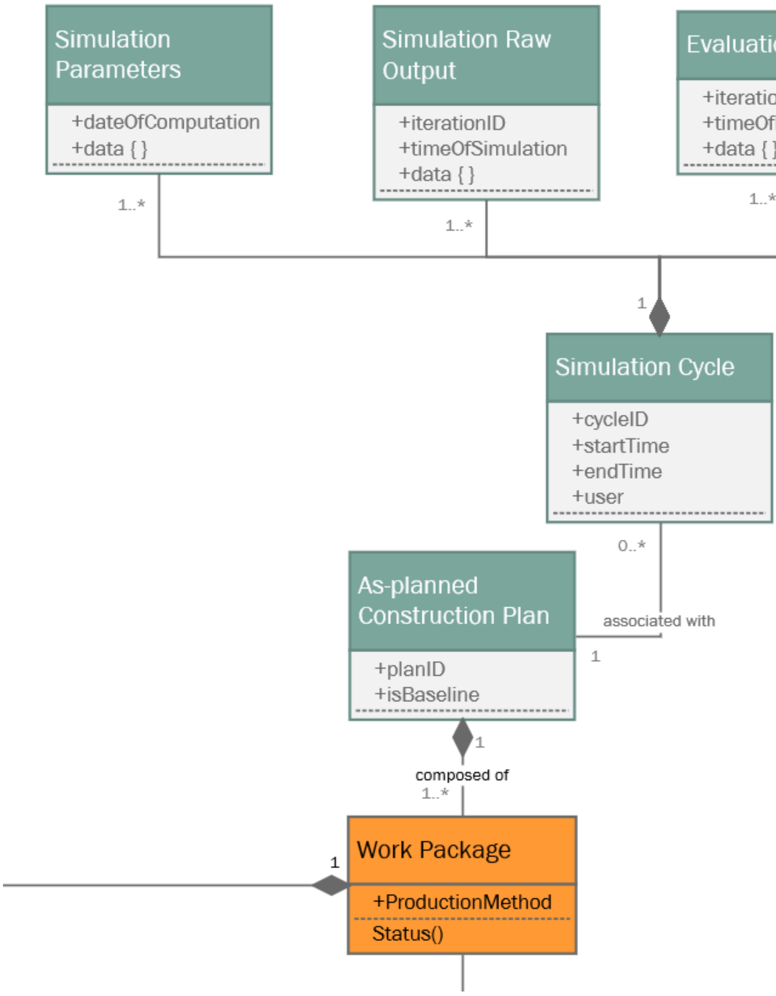


**Figure 49. UML class diagram for simulation input and output**

The as-performed process model (Figure 50) represents the as-performed processes captured from the construction site. This information belongs to the Project Status Information (PSI) and is encapsulated in the PSK. The proposed model parallels the as-planned process model (Figure 47) in its data structure, highlighting the intricate physical-virtual and status-intent relationships in DTC.

The core process components of the as-performed model are Construction, Operation, and Actions, with each of the former being a composition of the latter. Construction and Operation are essentially the same in nature as the Work Package and Activity respectively on the as-planned side. Where execution follows plan perfectly, their instances will be directly equivalent and have the same property values and relationships. However, in most cases, execution does not follow plan precisely, so that deviations from plan will arise.

Small deviations may require m-n relationships between instances of Activities (as-planned) and instances of Operations (as-performed). More significant deviations will lead to m-n relationships between Work Package instances (as-planned) and Construction instances (as-performed). The Action class instances, however, are expected to correspond in form and association with the Task instances, although they may belong to non-corresponding Activity/Operation instances.

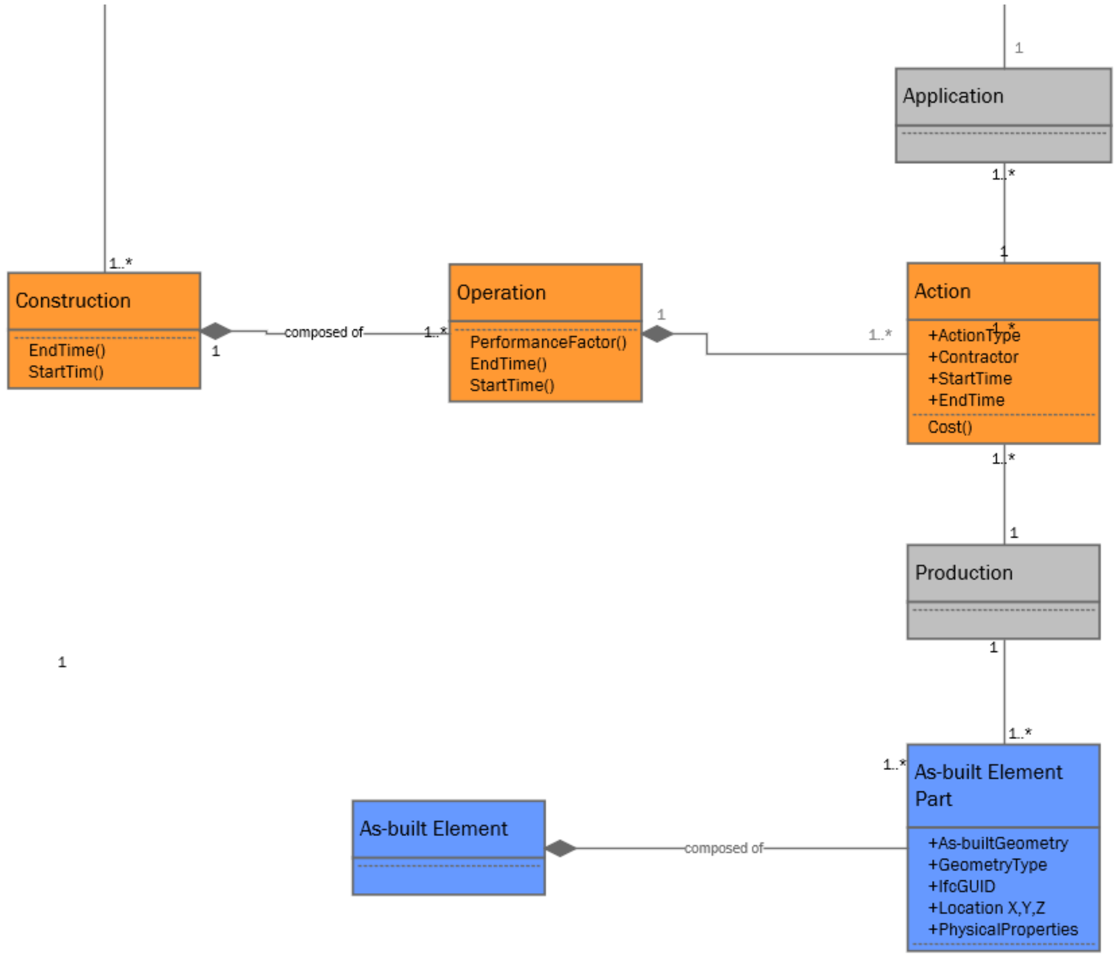


**Figure 50. UML class diagram for as-performed process model**

The As-performed Zones reflect the on-site spatial organization and its characteristics as they were actually performed. The data structure is identical to that in the as-planned model. The key difference that differentiates them from their as-planned counterpart is that they are expected to change dynamically daily or hourly to reflect the actual situation on site.

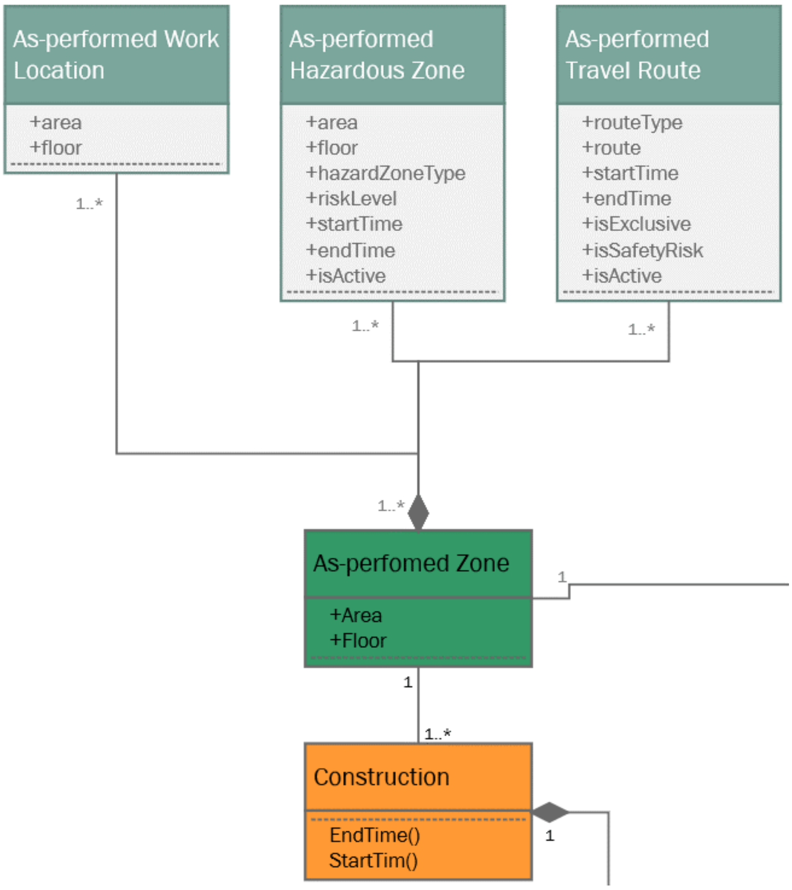


**Figure 51. UML class diagram for as-performed zone**

The data schema presented in Figure 52 details how the as-performed resource application can be represented in the DBTP. The As-performed Resources maintain the same class structure as their as-planned counterparts, with the main difference being that the equipment maintenance and material delivery classes represent specific instances of such events rather than the entire planned schedule. The As-Performed Resources in the PSI are also associated with Performance Indicators residing in the PSK.

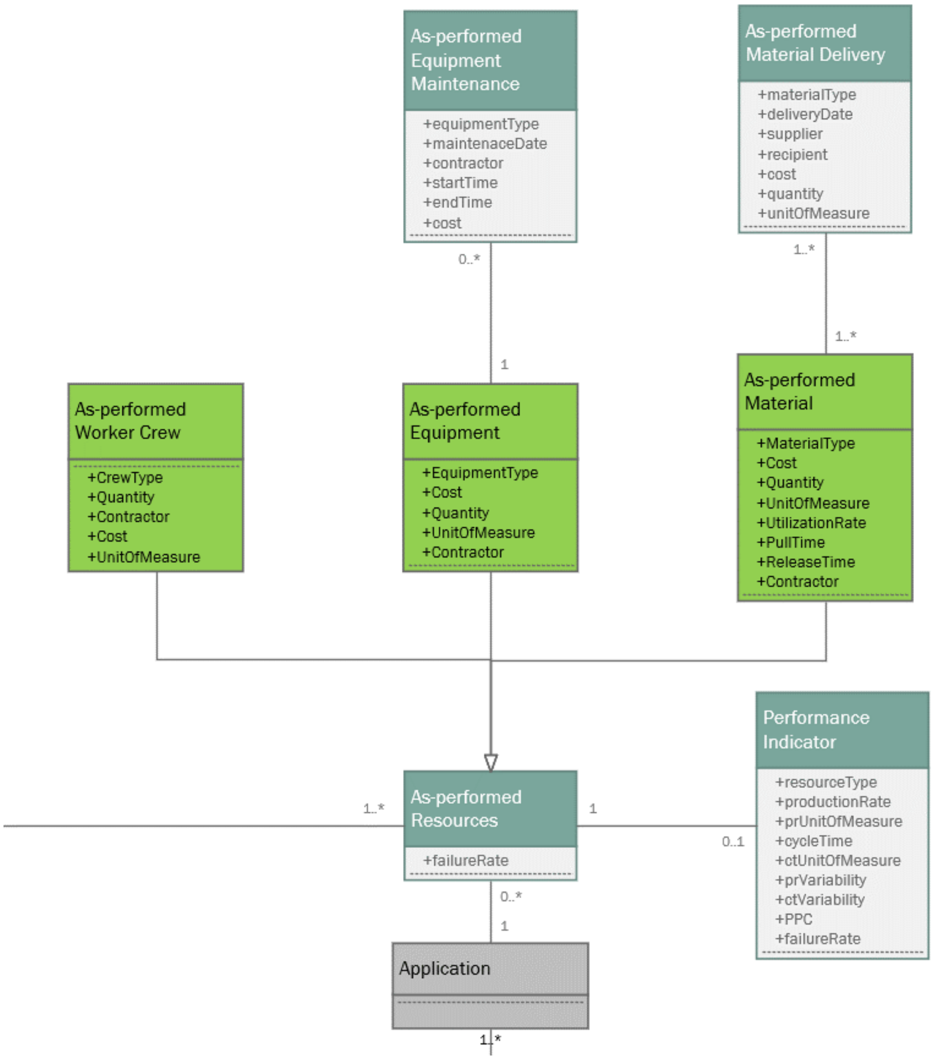


**Figure 52. UML class diagram for as-performed resource application**

### 6.7 Sensor network modelling

BIM2TWIN approach turns the construction site into a smart factory that is aware of its own state and aware of the work progression. A smart factory requires the installation new digital equipment such as sensors, cameras, lasers, digital network and servers. Those equipment belong to the construction site and should considered while describing the as-planned processes (digital equipment installation tasks) and as-design site (digital network description). This part of the model is common to various use case and should generalized. We suggest to implement the SSN/SOSA ontology to describe sensors, their location, their link with monitored objects (sosa:hasFeatureOfInterest, sosa:ObservesProperty), their link with set of data collected (sosa:hasResult), and their connection with the digital system (sosa:isHostedBy). In the following diagram we linked the SSN/SOSA approach with the equipment class of the BIM2TWIN core model. The Equipment (e.g. a crane) is a feature of interest (sosa:featureOfInterest), its properties (e.g. the its orientation) are observed (sosa:observedProperty). The observation (sosa:Observation) is made by a sensor (e.g. a camera).

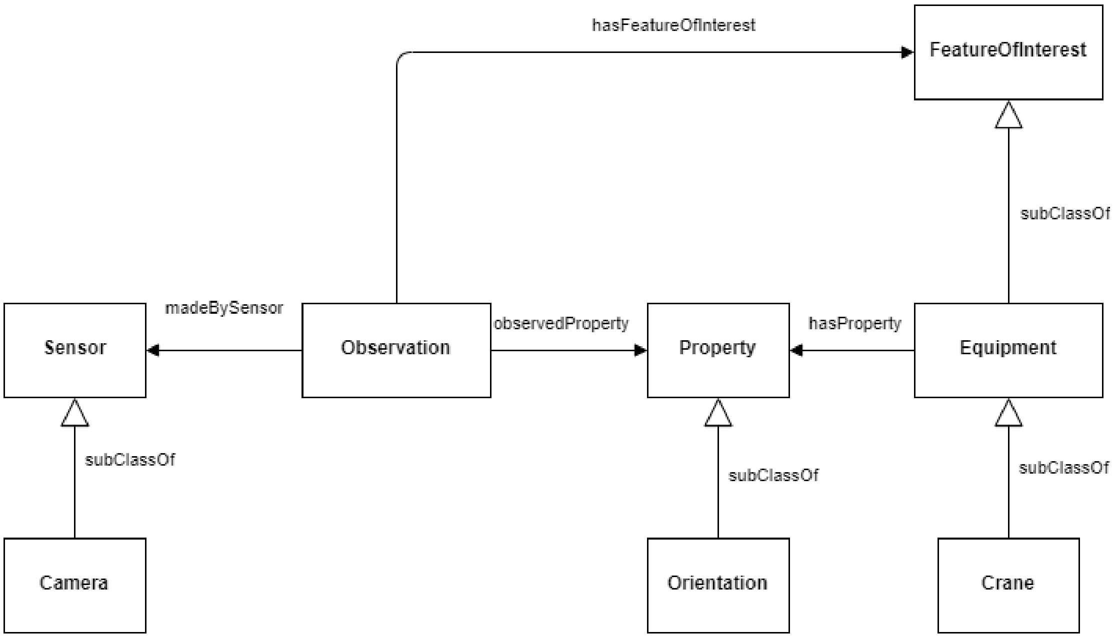


**Figure 53 - Sensor network model applied to BIM2TWIN**

## 7 IDENTIFYING DATA LAYERS

Data are organized in the graph in different layers. From the lower layer: raw data directly gathered from sensors and information sources to the upper layer KPIs. Layers are connected at the object level so that we can keep track of the data involved in KPIs calculation from sources to the final result. We have identified two kinds of data processes: real time processes (time step from second to hour time) that detect instant and local issues directly from raw data and midterm processes (time step from day to week) based on the logical layer and that keep trace of all events and allow to calculate KPIs and make strategic decisions. Information follows an ascending path from raw layer to layers with more value-added at each step. Each upper layer stores data one step closer to decision-making required information. At each step, data is filtered and aggregated.

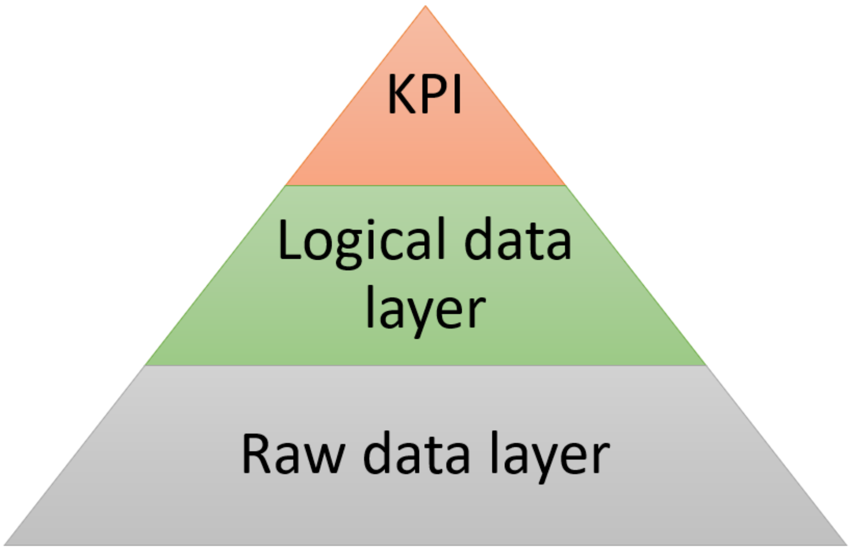


**Figure 54 - Layer organisation of the whole data model from raw data to KPIs**

The following table shows examples of data chain required to compute KPIs through each layer. To calculate Cost (KPI), we need the working duration and unit cost of each worker (Statistical layer), from the worker instances and history (Logical model), captured on-site by cameras as geolocated entities (Raw data).

| Raw data | Logical model | KPI |
|---|---|---|
| Worker geolocation | Worker, role, assigned task | Cost |
| Point cloud | Slabs, walls, roof, volume as designed, volu | Construction progress |
| Worker geolocation | Worker, role, safe areas | Number of safety warning |
| Equipement geolocation | Equipement, type, task, status | Equipement use rate |

**Figure 55 - Example of coresponding data in each layer**

### 7.1 Raw data layer

The raw data layer is the data lake of the digital twin. It stores data coming from automated (sensors) and manual data flow (manual entries) as they are produced. **Raw Data Content** can be simple values such as

RFID or files stored in a file system such as raw geometries (Point clouds, images, IFC geometry) or sensors generated data. This layer provides **Raw Data Context** as metadata that documents, classifies those data, and allow to link them with logical instances as data sources. **Raw Data Context** is part of the knowledge graph and could be clearly identifying by a named graph.

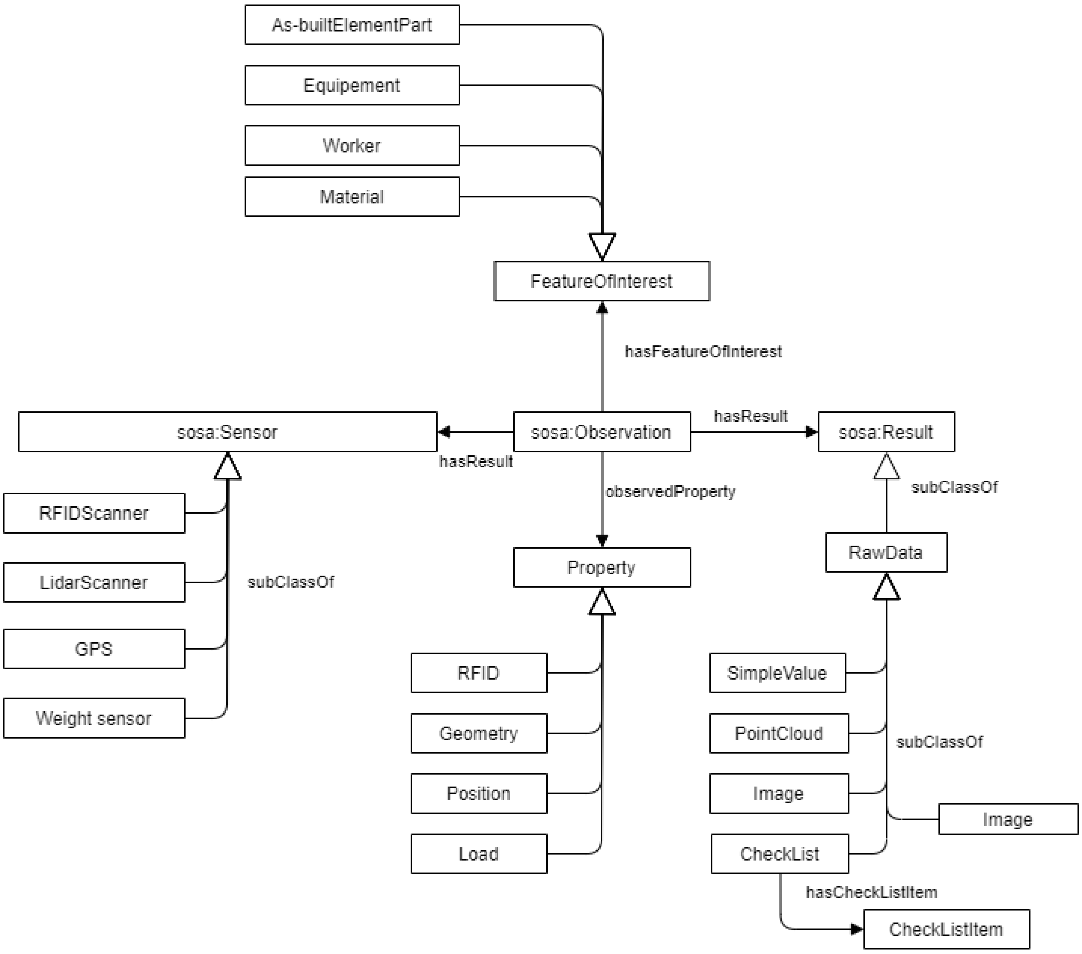


**Figure 56 - Raw Data Layer Model**

Level of details is driven by capture costs (sensors, capture algorithms licenses, human resources, quality control). Raw data can be partially processed locally before the be centralized in the Digital Twin.

All those data sets need to be referenced and described in the graph database with:

- **A unique identifier** to be distinct from all other datasets and instance in the graph database. The identifier could be a URI.
- **A name** that allows human recognition and understanding. The name could be the name of the source file or a concatenation of sensor identifier with timestamps.
- **A type** that classifies the type of data (cloud point, image, IFC model, measurement…)
- **A source:** sensor, human, software or machine that provided the data.
- **A resource location :** Link to raw data content such as HTTP served files located by an URL.
- **A location / geometry:** if the sensor is static (on site camera), the data location is given by the sensor location. If sensor is moving (camera on drone), the location should be recorded.

- **Historical record:** indicate when the data as be collected (timestamp).
- **Direct or indirect links with as-planned model and as-performed model**: we should be able to compare *as-planned* state and *as-performed state* for each object involved in KPIs. The state can be explicit (semantically link to the object in the graph), inferred (deduced from semantic neighbourhood by using logical statements or **rules**) or calculated (mostly when accurate geometry, time series or simulation algorithms are involved).

## 7.2 Logical data layer

Data from raw layer are processed, filtered, and organized in the logical model. This layer embeds atomic instances that mirror physical objects and persons. The level of detail of this layer is a trade-off driven by KPIs. The methodology and content of the logical layer is detailed in the following part.

## 7.3 KPI layer

KPI layers organize high level indicators that are displayed in the dashboard for actors to take right decisions.

Main identified KPIs are the following :

- For process and resources optimisation: Throughput, cycle time, order fulfilment, work capacity, workforce utilization rate and equipment use rate, overall and relative cost, overall and relative duration.
- For quality monitoring: Number of detects identified/fixed per work package, number of executions with no detects, number of rework, delays due to quality.
- For safety control: site safety level, defect solving times, risk assessment completion level, number of safety warning, accident frequency

These KPIS are defined in others deliverables of the B2T project : D1.3, D3.1, D4.1 and D5.1.

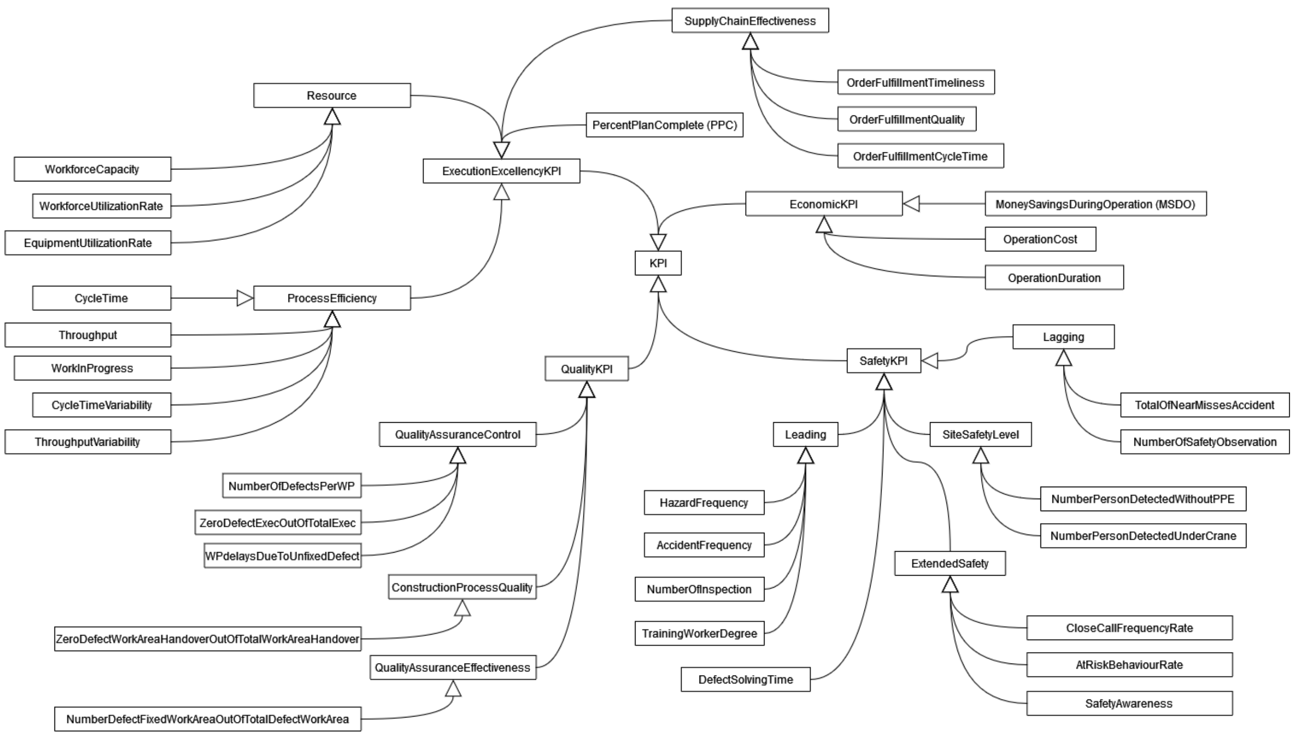


**Figure 57 - KPIs layer model**

# 8 Focus on The Logical Layer

In this part we focus on the logical layer as a central point of this document. In our analysis we adapted the level of detail according to KPIs calculation needs. Data coming from the Raw Data Layer such as sensors data flows are contextualized, filtered, aggregated, simplified to optimize statistical calculation and logical inferences. In the Logical Layer, the geometry is approximated to points or bounding boxes instead of exact tessellation. Time series are aggregated by adjusting time steps and by calculating average.

## 8.1 Identifying main classes

The logical layer integrates data models coming from the local requirement analysis. The integration work implies to identify and harmonize common concepts between local models while considering patterns provided by standards and existing ontologies.

The construction site can be seen as a factory where material entered is transformed into a building by people using machines. The main difference with traditional industry is that the factory -composed by scaffolding, formwork, cranes…- is built on site for during the project. Sub-task targets can be factory components (formwork, scaffolding) or material supply chain actions (delivery/reception, storage, transportation, transformation).

From an as-planned point of view

A **Task** **assigns persons**

**located in a specific site area**

**to transform materials into building components – that as specifications –**

**by using equipment,**

**along a timeline period**

**with respect to safety rules and process**

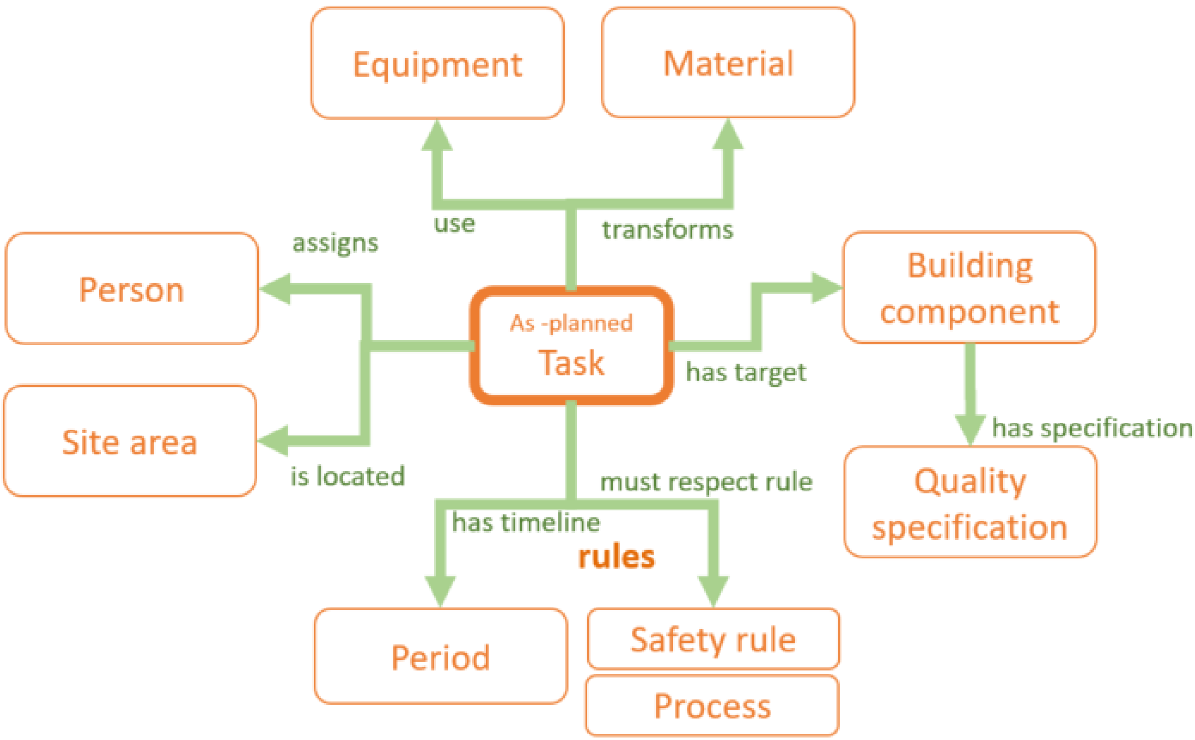


**Figure 58 – Overview of the as-planned integrated model**

From an as-performed point of view

A **Task** **assigned persons**

**located in a specific site area**

**and has transformed materials into building components that has a performed quality**

**and used equipment,**

**along a timeline period**

**following certain safety and process rules**

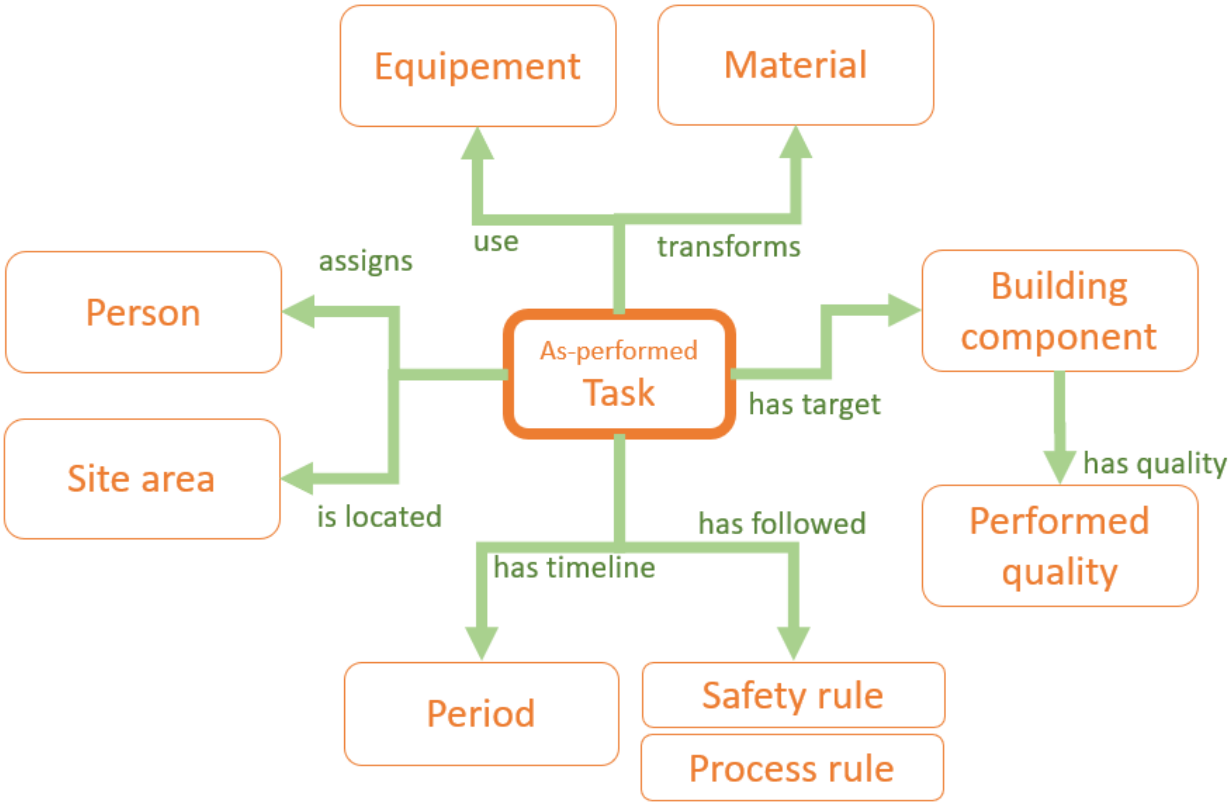


**Figure 59 - Overview of the as-performed model**

As soon as "as-planned" and "as-performed" tasks are -directly or indirectly- linked we can compare each aspects of the task to quality construction progress.

It is also possible to compare to different plans be comparing "as-planned" tasks of plan A to "as-planned' task of plan B, either task by task or in a global way.

In case a given performed quality doesn't reach the specifications then **a quality alert** must be thrown.
For a given task, if mandatory rules -processes and safety- are not followed then a **process alert** or **safety alert** must be thrown. If the period doesn't fit, a **delay alert** is thrown.
The cause of those alerts can be equipment (failure, unavailable…), person (unavailable, unsuitable skills), material (unavailable, unsuitable), site area (not accessible) or other conditions such as weather.

## 8.2 Identifying main attributes

The Five Ws (sometimes referred to as Five Ws and How, 5W1H, or Six Ws) are questions whose answers are considered basic in information gathering or problem-solving. They are often mentioned in journalism (cf. news style), research and police investigations.[2] According to the principle of the Five Ws, a report can only be considered complete if it answers these questions starting with an interrogative word:

**Who**: who is responsible for? Who to inform?

**What**: instance description with identifier, name, and type (or classification)

**When:** historical record of location and state

**Where:** location, orientation, topology (in area, out of area, closed to, touch) and geometry

**Why:** link with as-planned model as a target

We even can add a **How** to describe current (i.e. as -performed) state of instances

Each instance of the data (tasks, workers, materials, equipment, digital resources and building components) model need to be described with at least :

- **A unique identifier** to be distinct from other instances. Link Data provides the concept of URI[76] that guarantee unique identifiers in time and space at world scale.
- **A name** that allow human recognition and understanding. The name not required for data processing but will be use in user interface to enter and display data for human-to-machine interactions.
- **A type** that classify the object into a taxonomy of object type when the object class is not accurate enough.
- **Responsibility assignment:** indicate who is responsible for quality and use and who should be informed of the state of the object.
- **A location / geometry:** tasks, workers, materials, equipment's and building components should be geolocated. The location can be stored according to are various level of details: with a 3D point (and a tolerance), with a bounding box, with an oriented bounding box or a complex geometry (group of facets). As complex geometry is difficult to handle in a graph database, the geometry should be simplified as much as possible and raw geometry should be stored as binary resources linked to objects but stored as blobs or outside of the graph database. The level of detail depends on the object type and the accuracy required by use cases and KPIs
- **Historical records:** State, location and geometry should be historically recorded to be able to handle trajectories and path length. Tasks should be timestamped by their start and end date and time. Status of workers, materials, equipment's and building components should be also historized and located on the project life cycle timeline.
- **Direct or indirect links with as-planned model and as-performed model**: we should be able to compare as-planned state and as-performed state for each object involved in KPIs. The state can be explicit (semantically link to the object in the graph), inferred (deduced from semantic neighbourhood by using logical statements or **rules**) or calculated (Mostly when accurate geometry, time series or simulation algorithms are involved).
- **Safety rules**: each physical object as to be linked to **safety rules** to respect.

**Table 12 - Main concepts contextualization 1/2**

| | | | |
|---|---|---|---|
| Task | What | **Identifier** | UID |
| | | **Name** | [Type]-[Area]-[timestamp] |
| | | **Type** | Dictionnary taxonomy |
| | Who | **Responsibility** | Operational manager |
| | Where | **Location, geometry** | Point, sphere (diameter), attached to building component or equipment |
| | When | **Historical records** | Associated resources, state |
| | Why | **Link with as-planned** | Planification |
| | How | **Link with as-performed** | Current state |
| | | **Life cycle** | Dictionnary taxonomy |
| | | **Others** | |
| Worker | What | **Identifier** | Registration number |
| | | **Name** | Real name |
| | | **Type** | Dictionnary taxonomy |
| | Who | **Responsibility** | HR manager |
| | Where | **Location, geometry** | Point, cylinder (2mx1m) |
| | When | **Historical records** | Position, task assigment, state |
| | Why | **Link with as-planned** | Planification |
| | How | **Link with as-performed** | Current state |
| | | **Life cycle** | absent, present, staffed, unstaffed |
| | | **Others** | |
| Building components | What | **Identifier** | IFC GUID |
| | | **Name** | IFC name |
| | | **Type** | Dictionnary taxonomy |
| | Who | **Responsibility** | Work package manager |
| | Where | **Location, geometry** | Point, Bounding box, topological relation with other components |
| | When | **Historical records** | State (as-performed) |
| | Why | **Link with as-planned** | IFC model |
| | How | **Link with as-performed** | Current state |
| | | **Life cycle** | |
| | | **Others** | |

**Table 13 - Main concepts contextualization 2/2**

| | | | |
|---|---|---|---|
| Digital resource | What | **Identifier** | URL |
| | | **Name** | [Type]-[sensor]-[timestamp] |
| | | **Type** | Dictionnary taxonomy |
| | Who | **Responsibility** | Data manager |
| | Where | **Location, geometry** | Sensor location when collected, URL |
| | When | **Historical records** | Delivered, deprecated |
| | Why | **Link with as-planned** | Planification |
| | How | **Link with as-performed** | Presence, quality |
| | | **Life cycle** | |
| | | **Others** | quality |
| Material | What | **Identifier** | UID |
| | | **Name** | [Type]-[delivery date]-[timestamp] |
| | | **Type** | Dictionnary taxonomy |
| | Who | **Responsibility** | Delivery manager |
| | Where | **Location, geometry** | Point, Volume (quantity) |
| | When | **Historical records** | Position, state |
| | Why | **Link with as-planned** | |
| | How | **Link with as-performed** | |
| | | **Life cycle** | delivered, stored, positionned, in use, final state / waste |
| | | **Others** | Providers, delivery date, transformation date, previous state, quantity |
| Equipment | What | **Identifier** | Registration number |
| | | **Name** | [Type]-[registration number] |
| | | **Type** | Dictionnary taxonomy |
| | Who | **Responsibility** | Technical manager |
| | Where | **Location, geometry** | Point, Bounding box, Orientation |
| | When | **Historical records** | Location, state |
| | Why | **Link with as-planned** | |
| | How | **Link with as-performed** | |
| | | **Life cycle** | stored/unused, in use, out of work |
| | | **Others** | |

## 8.3 Federating and aligning with existing ontologies

As presented in part 4, there are three possible strategies for integrating industry-standard ontologies for a specific domain like digital twin for construction site: federate, align or customize. Those three strategies can be mixed up and the strategy choice can concern only a part of the whole digital twin model.

Two independently developed ontologies can exist with (partially) similar goals. To make these ontologies and the data described with it interoperable, the terminology of both ontologies should be related to each other through an ontology alignment process. When two ontologies are complementary, they can be linked through a common or aligned concept or by creating a new relation between two concepts from each ontology.

We have analysed four high level ontologies covering the construction domain. The four ontologies CTO, DiCon, ConTax and BOT have been presented in part 5. The next table is a matrix that shows the connection between thoses four ontologies and others existing ontologies and the alignment strategy applied. OWL, RDF, RDFS and SKOS are used by the four ontologies as form the web semantic kernel.

**Table 14 - Examples of alignment strategies**

| | CTO | DiCon | ConTax | BOT |
|---|---|---|---|---|
| ConTax | Federate | | Self | |
| ConTask | Self | | Federate | |
| DiCon | | Self | | |
| BOT | Federate | Align | Federate | |
| Dublin Core | Federate | | Federate | Federate |
| Damage Topology Ontology | Federate | | Federate | |
| OWL | Federate | Federate | Federate | Federate |
| RDF | Federate | Federate | Federate | Federate |
| RDFS | Federate | Federate | Federate | Federate |
| Schema.org | | | Federate | Federate |
| VANN | Federate | | Federate | Federate |
| VCARD | Federate | | Federate | |
| VOAF | Federate | Federate | Federate | Federate |
| XSD | Federate | Federate | Federate | Federate |
| SKOS | Federate | Federate | Federate | Federate |
| BFO | | Align | | |
| QUDT | | Align | | |
| OPM | | Align | | |
| wgs84_pos | | Align | | |
| ifcOWL | | Align | | Align |
| OWL-Time | | Align | | |
| FOAF | | Align | | |
| Org | | Align | | |
| PROV-O | | Align | | |
| SSN | | Align | | |
| SOSA | | Align | | |
| DCAT | | | Federate | |
| FOG | | | Federate | |
| OMG | | | Federate | |
| OPM | | | Federate | |

It appears that CTO, ConTax and BOT are linked together and mainly adopt the federation strategy whereas the DiCon seems isolated from the three others but is align with many other well know ontologies. BOT is aligned with ifcOWL as it appears to be a simplification of IFC concepts.

**Table 15 - Sample of alignment between IFC and BOT**

| Subject | Predicate | Object |
|---|---|---|
| ifc:IfcSite | rdfs:subClassOf | bot:Site |
| ifc:IfcBuilding | rdfs:subClassOf | bot:Building |
| ifc:IfcBuildingStorey | rdfs:subClassOf | bot:Storey |
| ifc:IfcSpace | rdfs:subClassOf | bot:Space |
| ifc:IfcElement | rdfs:subClassOf | bot:Element |

DiCon is aligned with sensor network description ontologies (SSN/SOSA) that are a significant part of the Digital Twin concept.

**Table 16 - Sample of alignment between DiCon and SOSA**

| | | |
|---|---|---|
| Dicv:Subject | owl:equivalentClass | sosa:FeatureOfInterest . |
| dicv:Property | owl:equivalentClass | ssn:Property . |
| sosa:ObservableProperty | rdfs:subClassOf | dicv:Property . |
| dicv:QuantitativeState | owl:equivalentClass | sosa:Result . |
| dice:Sensor | owl:equivalentClass | sosa:Sensor . |
| ssn:Platform | rdfs:subClassOf | dice:MaterialEntity . |
| dicp:Observation | owl:equivalentClass | sosa:Observation . |
| dicp:Actuation | owl:equivalentClass | sosa:Actuation . |
| sosa:observedProperty | rdfs:subPropertyOf | dicp:hasObservedProperty . |
| sosa:actsOnProperty | rdfs:subPropertyOf | dicp:actsOnProperty . |
| sosa:hasResult | owl:equivalentProperty | dicp:hasObservedResult |

DiCon is also aligned with QUDT, OWL-Time, Prov-O ontologies that are widely used in the engineering domain to describes process and physical properties.

BIM2TWIN approaches can federates various domain specific ontologies around a core model defining main as-planned and as-performed models.

The following table makes a recap of ontologies that fit or partially fit with the BIM2TWIN scope. For each ontology, we mention the BIM2TWIN domain of application, the level -low level is close to technical data management while high level is close to business use cases-, the main concepts in the kernel of the ontology and the concepts from BIM2TWIN we could link or align with.

**Table 17 - Ontologies in scope of BIM2TWIN**

| Ontology | Domain | Related B2T Concepts |
|---|---|---|
| INPIRES ontologies | Urban Digital Twin | Building, Site |
| BOT | As-designed Building topology | Building, BuildingElement |
| geoSPARQL | As-designed / as-built geometry | Geometry, bounding boxes |
| BEO | As-designed Building components | Product, BuildingElement |
| Omniclass taxonomy | As-designed Building components | Product, Building Element |
| SSN/SOSA | Sensor network | Building Element, Resource, Equipement, Sensor |
| Core Security | Safety | Person, Assests, Security Requirement |
| Damage Topology | As -built Quality | Defect, Building element |
| Digital Construction | As-planned processes | Activity, precondition |
| Suppply Chain Ontolog | Processes | Material, Role, Resource |
| OWL-time | Processes | Acitivity, RawData |
| PROV ontology | As-performed Processes (Data), raw data layer | Agent, Activity, Resource, Equipement, Data source |
| SKOS | Taxonomies, alignments | All concepts with taxonomies |
| QUDT | Quality, Material quantities, raw data layer, KPIs la | All physical and theorical quantities with units |
| Dublin Core | Raw data layer | Raw dataset |
| KPI Ontology | KPIs layer | KPI |

Although we found various research work on Linked Data oriented Building Digital Twin models, the construction site digital twin seems to be a newer topic of study. Some ontologies can be shared with the Building Digital Twin approaches such as Building Model (BOT), and sensor networks (SSN/SOSA).

We found that various low-level ontologies such as SKOS, PROV, OWL-TIME, QUDT provide a good framework on top of which we can build more specific vocabulary.

In the Logical Layer graph, simplified geometry can be described by using the Well Known Text Coordinates Representation (WKT)[77] as literals.

After the first analyse, we can make a projection of those ontologies on the BIM2TWIN domains to show the covering rate. This projection shows that low level concepts get the best covering followed by the as-planned core model. Meanwhile, as-planned quality and whole equipment domain miss alignment with existing ontologies.

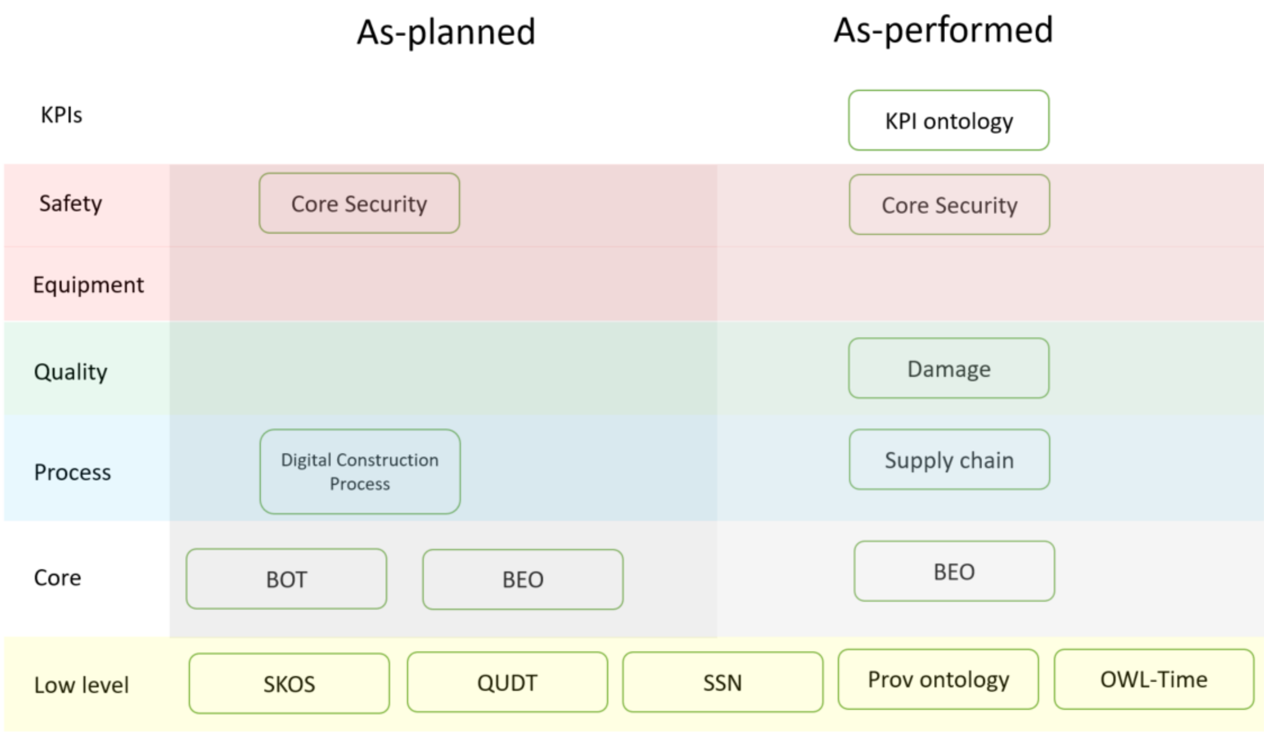


**Figure 60 - Overlap between exiting ontologies and BIM2TWIN scope**

This lack of alignment could be fixed, if needed by studying a wide range of ontologies including less stables (experimental) ontologies or draft research models. In the meantime, we will defines ad'hoc concepts for specific domains that could be aligned afterwards. The following table shows the alignment strategy suggested for each ontology candidate as previously defined in part 5.

**Table 18 - - Relevant ontology and related alignment strategy**

| Ontology | Domain | Related B2T Concepts | Alignment strategy |
|---|---|---|---|
| INPIRES ontologies | Urban Digital Twin | Building, Site | Align |
| BOT | As-designed Building topology | Building, BuildingElement | Federate |
| geoSPARQL | As-designed / as-built geometry | Geometry, bounding boxes | Federate |
| BEO | As-designed Building components | Product, BuildingElement | Federate |
| Omniclass taxonomy | As-designed Building components | Product, Building Element | Align |
| SSN/SOSA | Sensor network | Building Element, Resource, Equipement, Sensor | Federate |
| Core Security | Safety | Person, Assests, Security Requirement | Custom |
| Damage Topology | As -built Quality | Defect, Building element | Custom |
| Digital Construction | As-planned processes | Activity, precondition | Align |
| Suppply Chain Ontolog | Processes | Material, Role, Resource | Align |
| OWL-time | Processes | Acitivity, RawData | Federate |
| PROV ontology | As-performed Processes (Data), raw data layer | Agent, Activity, Resource, Equipement, Data source | Federate |
| SKOS | Taxonomies, alignments | All concepts with taxonomies | Federate |
| QUDT | Quality, Material quantities, raw data layer, KPIs la | All physical and theorical quantities with units | Federate |
| Dublin Core | Raw data layer | Raw dataset | Federate |
| KPI Ontology | KPIs layer | KPI | Align |

## 8.4 Deriving the BIM2TWIN ontology

In the previous, we have identified overlapping concepts with existing ontology and the preferred method to link them with the BIM2TWIN core ontology. For ontologies that are federated, we identified the commun concept that articulates them with BIM2TWIN ontology. **BOT** is used for Building design. It is connected to the BIM2TWIN ontolgy through the Building **Element** concept. The **Element** concept is contextualized as **As-design Element** or **As-perform Element** whether it is an target or an actual production. As seen in the Raw Data Layer, Building **Element** can be a **Feature of Interest** (as definied in **SOSA**) with properties observed by a sensor. Provenance of **Raw Data** (as definied in the Raw Data Layer) is partialy contextualized through **SOSA** (as an kind of observation result) and will be enriched with the use of **PROV-O** ontology. Temporal contextualization of **Actions**, **Tasks** and **Observations** are modelled according to **OWL-TIME** ontology. Building **Element** classification is definied by the **BEO** taxonomy, other classifications and taxonomies (equipement, taks, issues…) are defined by using **SKOS** ontology in the BIM2TWIN dictionnary. Physical quantities such as those observed by sensors or in KPIs (e.g. amount of materials) are definied by using **QUDT** unit classification.

The following diagrams presents in details this core ontology derived from of the core model, the domain model.

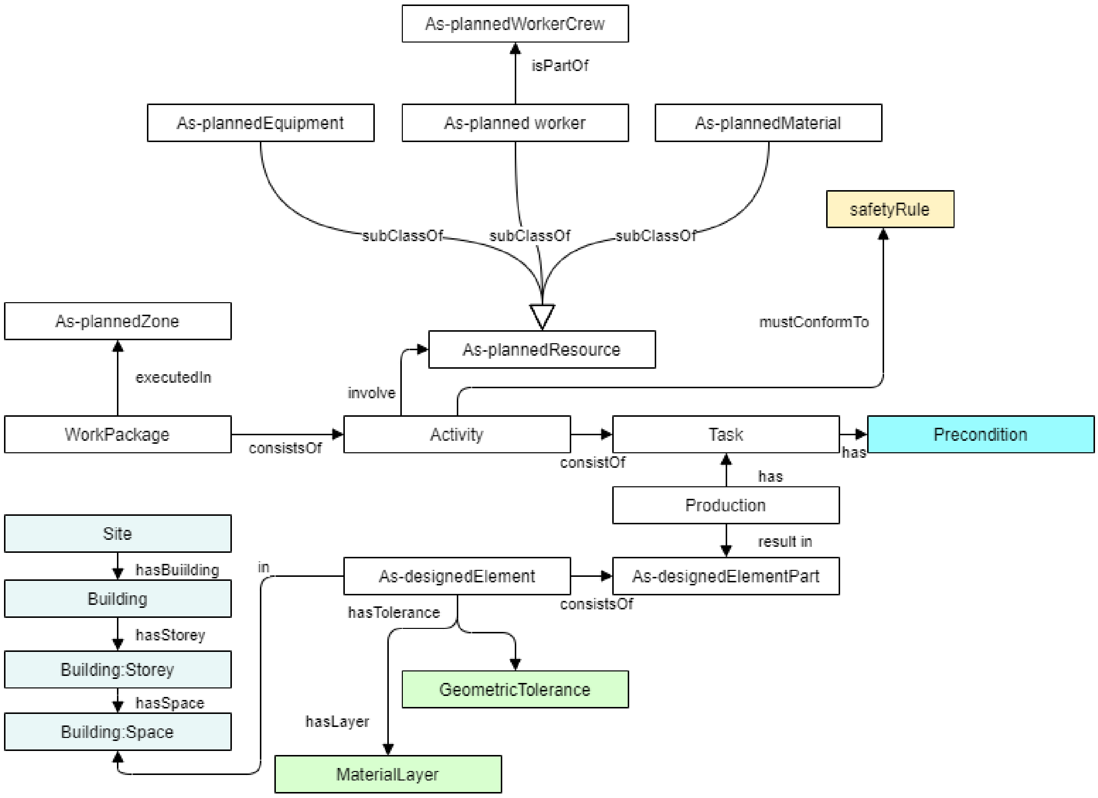


**Figure 61 - BIM2TWIN ontology, As-planned part**

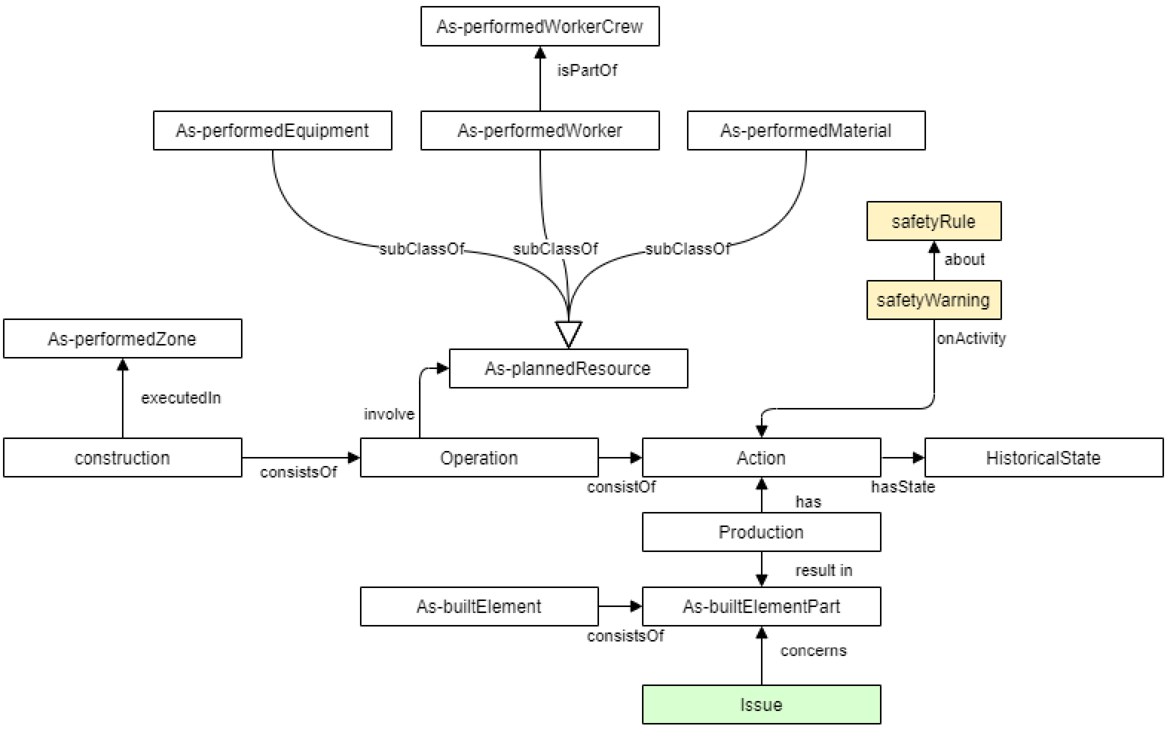


**Figure 62 - BIM2TWIN ontology, As-performed part**

Diffent kind of **Issue** are classified in a taxonomy. Different subclasses of **Issue** can have different signature depending on the context. An issue can only concern a **As-planned Task** when the task has not been realized at all. An issue can also connect a **As-design Element** with a **As-built Element** when the element produced by an (as-performed) **Action** fits with the (as-planned) **Task** but doesn't not meet the quality requierments defined at design phase.

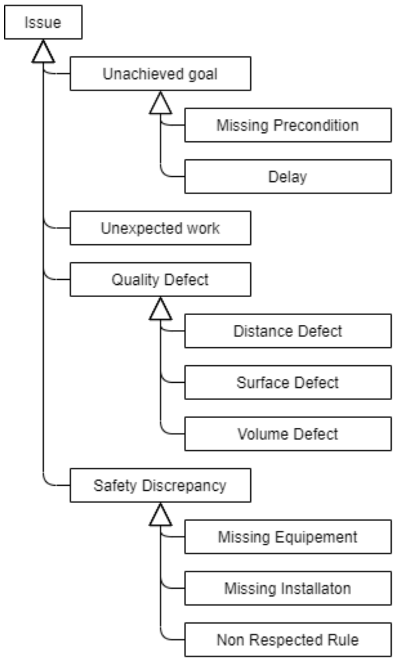


**Figure 63 - Issues taxonomy**

**Unachieved Goal** is a kind of issue that defines a planned **Task** that as not been completed on time. **Unachived-Goal** can be refined into subclasses such as **Missing Precondition** or **Delay** that can implements a higher level of detail.

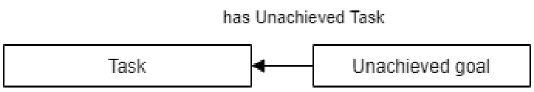


**Figure 64 - Case of an unachived goal**

**Quality Defect** model gaps between actual quality produced and expected quality. A **Quality Defect** instance is linked to the as-designed target through the class **As-designed Element** and linked to the as-built model through the class **As-built Element.** Higher level of detail can be implement by using subclasses of **Quality Defect** such as **Distance Defect, Volume Defect or Surface Defect . Quality defect** is computed by the **Progress processor** that will compare the actual geometry with the required geometry.

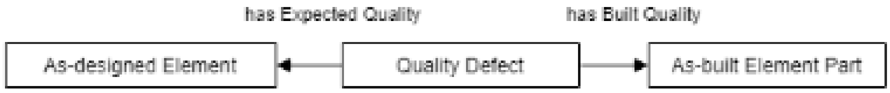


**Figure 65 - Case of a quality defect**

When an Element produced can not be linked to a designed element or a part of a designed element, it is considered as **Unexpected Work** .

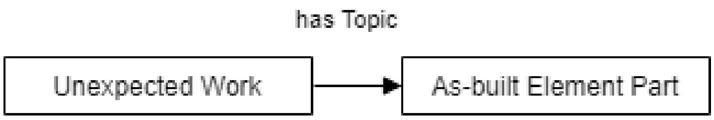


**Figure 66 - Case of on unexpected work**

A **Safety Discrepancy** result of a performed action that infridge a **Safety Rule**. Safety Discrepancy could be caused be a **Missing Installation** (e.g. missing guardrail) or a **Missing Equipement** (e.g. Hard hat)

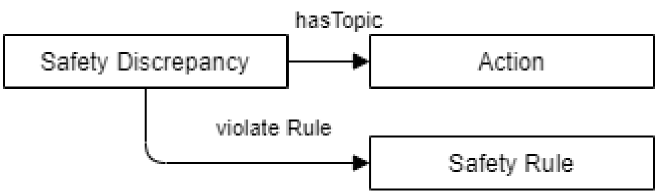


**Figure 67 - Case of a safety discrepancy**

When a performed **Action** fits with the actual plan the **Progress Processor** calculates a degree of progress that result in a **Progress** instance linking the Action with a planned **Task**.


**Figure 68 - Case of a normal progress**

From an **Abox** (Intances domain) point of view, **Progress** and various **Issue** instances can be cumulated. As an example, the following diagram represents an **Action** (#45) that has realized a planned **Task** (#85) at 85%, but that has broken the **Hard Hat Rule** (workers must wear hard hats when in working zones) and causes an axis deviation in comparison to the expected geometry.

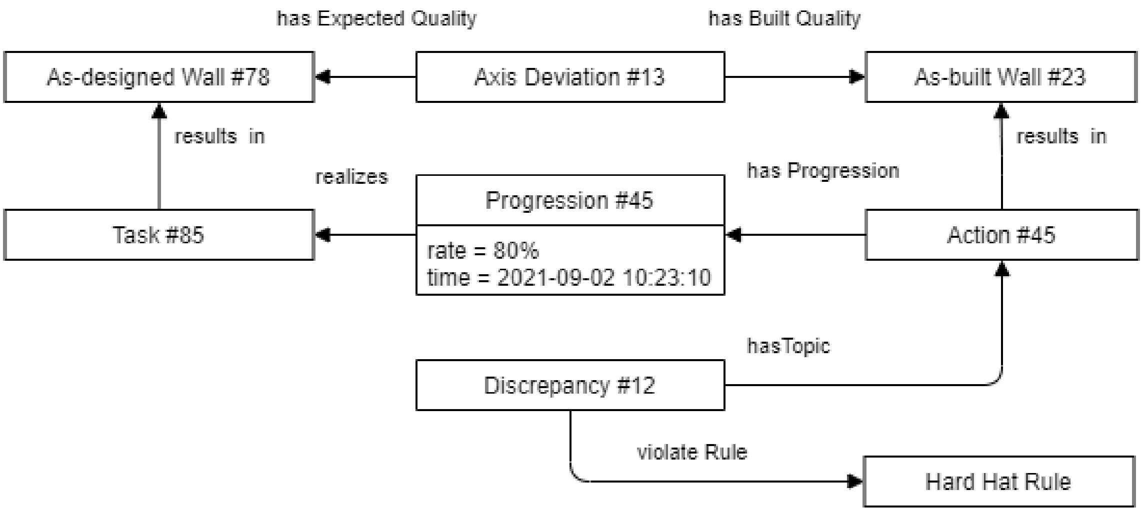


**Figure 69 - Issue model instanciation**

The next part describes the data workflow. In this workflow a data processor called "Data Aggregation Processor" will be in charge of computing KPIs. Most KPIs regarding **Quality**, **Safety** and **Progress** will be processed by aggregating **Issues** and **Progress** instances of the Logical Layer.

# 9 REQUIRED DATA PROCESSES

We have identified two kinds of data processes: real time processes (time step from second to hour time) that detect instant and local issues directly from raw data and midterm processes (time step from day to week) based on the logical layer that keep trace of all events, allow to calculate KPIs, and making strategic decisions. In this document we mainly focus on the logical layer and midterm processes but provide connections with the lower and uppers layers.

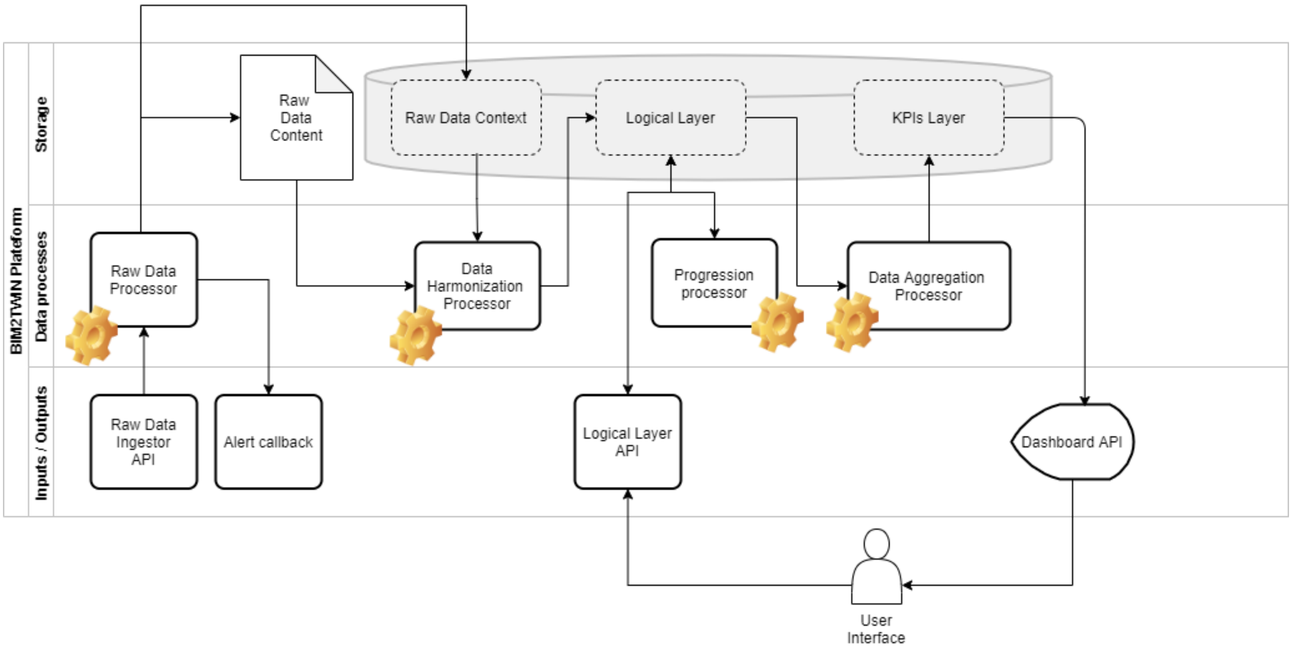


**Figure 70 - General data workflow**

In previous diagram, the three layers or the **Storage Lane** can be implemented as three named graphs[78] (e.g. https://bim4ren.eu/project/123e4567-e89b-12d3-a456-426652340000/raw-layer/) in the same knowledge database. Named graphs can also be used to identify alternatives plans (e.g. https://bim4ren.eu/project/123e4567-e89b-12d3-a456-426652340000/plan/1523/). The raw data layer is mainly implemented by a file system with resources identified by URL and contextualized through metadata from a named graph. Logical layer entities are linked to raw datasets to provide traceability. KPIs evaluations are connected to Logical layer entities involved in calculation.

The **Data Process Lane** provides an event manager are message bus to notify internal or external processes when data are updated.

- **Raw Data Processor** : is in charge of loading, identifying, contextualizing (metadata on provenance, timestamps, usage) Raw Datasets. Raw Data Processor analyses incoming Raw Datasets for critical and real time alert launches. Regarding platform capabilities and performance goals, as an alternative architecture, we could have some data pre-processed locally by the sensor network. As mentioned by Petar Kochovski and al while introduction the concepts of edge computing, “high Quality of Service (QoS) can be achieved by using adequate number and quality of computing resources, such as processing, memory and networking elements, geographically close to the smart environments”.

- **Data Harmonization Processor** :  Process incoming raw datasets and update logical layer. The Data Harmonization Processor is in charge of aligning data to the BIM2TWIN ontology with the expected level of details. Link logical entities and related raw dataset for traceability.
- **Progress Processor** : The progression processor regularly compare as-built against as-designed elements, as well as as as-performed against as-planned activities. This comparison results in updated links between target side (as-designed, as-planned) and real progression side (as-built, as-performed) and computed inspection results. Inspection on results can be Achievement States or Issues. When as-built elements don’t meet as-design requirements Quality **Issues** are produced. Non-planned activities and delays results in Plan **Issues**.  The Progression Processor also execute Safety Rules to produce Safety **Issues**.
- **Data Aggregation Processor** : Refresh KPI layer on a regular base or triggered by logical layer updates. This processor is mainly based on filter, aggregation and statistical calculation on issues, resource and activity state.

The **Input / Output lane** is a prefiguration of the API. The raw data collector allow real-time data providers as sensors network to connect to the Digital Twin Knowledge Platform and to upload data through specialized injectors (or connectors). We identified three kinds of injectors. Cold **data injectors** allow to initiate knowledge graphs with long term information such as as-designed target building, activity plan at project start or sensor network. **Hot data injectors** collect data as stream or time series that are buffered in the raw data layer.

# 10 Platform requirements

The following requirement are provided to ensure the correct implementation of schemas and knowledge graph quality. Those requirements are mainly based on W3C recommendation regarding web interfaces, Semantic Web Technologies and Linked Data principles.

## 10.1 Model implementation and Data quality

**RQ101** : Instances are identified by Universal Resource Identifiers (URI) in the knowledge graph.

**RQ102** : The platform provides functionalities to validate input data against expected data schema

**RQ103** : Data validation rules are formalized with a standard rule language such as SHACL (Shapes Constraint Language) or equivalent.

**RQ104** : The platform annotates raw data with context information according to metadata defined in the model particularly for non RDF data such as images, time series or point clouds

**RQ105** : Data are indexed when stored to provide read performances (metadata, geospatial indexes, temporal indexes)

**RQ106**: Geometries are described as literals embedding non-RDF geometry as defined by the Well Known Text ontology (WKT)

**RQ107** : Data schemas are documented and published on the platform. Model introspection allow the system to access and infer on the documentation.

**RQ108** : The knowledge graph implements named graphs in order to identity datasets within a project such as layers or **Alternative Plans**. Named graphs have identifiers, they can be associated with relevant metadata to describe context, such as provenance information or status.

**RQ109** : The platform provides versioning functionalities. In the Raw Data Layer, SOSA approach, should be sufficient to timestamp data and keep history. In the Logical Layer, Construction plans and as-build progression need to be versioned. This versioning could be achieved by using named graph.

## 10.2 Storage

**RQ201**: The platform stores raw data as files or times series.

**RQ202**: Data can be stored in named graphs to organize data layers and data versioning (Cf. RQ 108)

**RQ203**: Data can be imported, exported according to Linked Data standards (Such as RDF, Turtle and JSON-LD)

**RQ204**: Raw dataset stored as files are identified by URLs and contextuatlized by metadata (as JSON-LD files or in a named graph).

## 10.3 Process

**RQ301**: Events are raised through a message bus when data are ready to be used

**RQ302**: Internal or external can subscribe to events to be notified

**RQ303** : The data provides ETL and Conversion to RDF capabilities. A mapping language between non-RDF and RDF is implemented such as SPARQL-Generate. As part of the Data Harmonization Processor, the platform implements injectors to convert data standard formats to RDF graphs.

## 10.4 Graph requesting and updating

**RQ401** : Graph requesting is provided through a standard language compliant with RDF graphs such as SPARQL.

**RQ402** : OWL is implemented to provide inference on concept extensions relations such as sub-classes and sub-properties.

**RQ403** : The system provides inferences capabilities on geospatial data (OGC topological relations), pattern recognition, graph traversal. From one perspective, generic reasoning based on logics defined with RDFS and/or OWL terminology is very useful since it allows to infer implicit triples.

**RQ404**: Request language provides aggregation functions to perform statistical calculations on groups of data (e.g. SUM, AVERAGE, MIN, MAX…).

**RQ405**: Update queries use the same core syntax as request (such as SPARQL INSERT )

**RQ406**: The platform provides a Standard API (e.g. SPARQL ENDPOINT REST API) to request the knowledge graph.

## 10.5 Visualization

**RQ501**: The platform provides web HTML5 (CSS, HTML, SVG, JavaScript) compliance to display user interface such as interactive dashboards.

**RQ502**: The platform provides functionalities to visualize 2D and 3D representations of the construction site with built elements and resources.

**RQ503**: Indicators of the dashboard should be connected to geometrical representations of entities involved in calculation.

**RQ504**: The user can interact with element to get states and details.

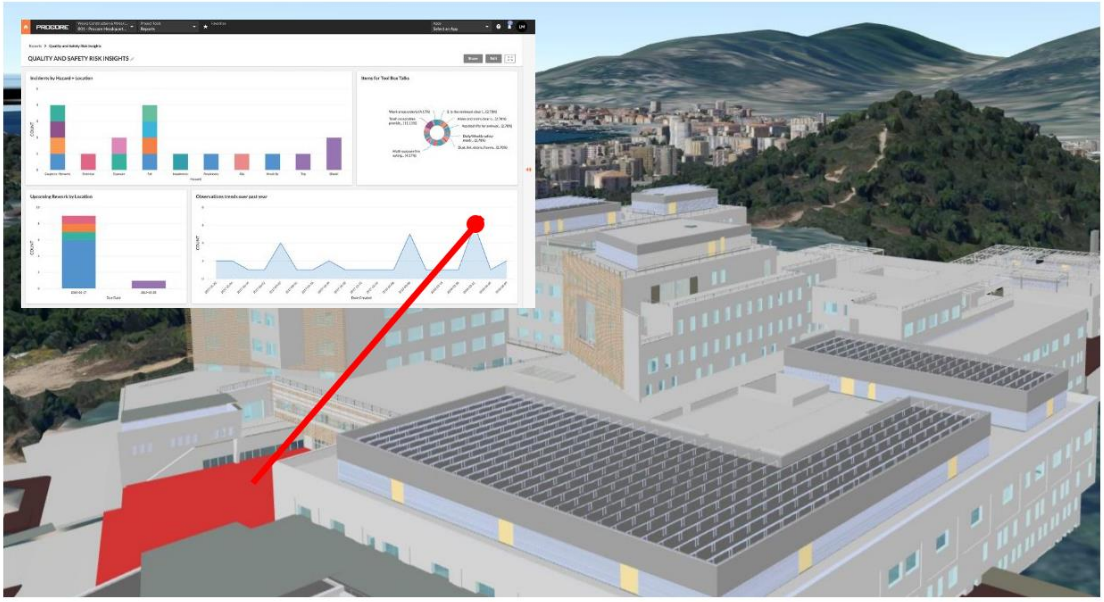

**Figure 71 - Dashboard prefiguration including KPIs display**

## 11 CONCLUSION AND NEXT WORKS

This report provides guidelines to implement, enrich and maintain the ontology of the Digital Construction Site Twin. It is based on research and industry standards to provide an optimal interoperability. Digital Twin Model is mainly graph-oriented respectful of W3C recommendations. It is organized in three different layers : Raw data Layer, Logical Layer, KPIs Layer. The Data Layer is format agnostic and should adapt to any source of data; complementary alignments should enrich this first work to develop injectors for proprietary data formats. The Logical Layer is expandable, it is built around a Core Model for domain specific concepts to be linked to. The KPI layer models indicators to provides high-value data for decision making. A general data workflow is provided to organize data process from raw data to KPIs.

This document also provides some requirements for the model implementation in the Digital Twin Platform. As a next step, together with Orange and other partners involved in WP2, we will implement core and federated ontologies in the Orange platform (WP2.6). Data exchange schemas as well as API methods needed to operate the platform will be derived from that model (WP2.7).

New releases of the ontology will be documented through the Dictionary Approach. The model provided will be tested through real data as they are collected by pilots. It will be adapted to the constraints brought by real data (availability, level of detail, accuracy, cost of capture, format of existing data). The BIM2TWIN should contribute to validate federated ontologies and to enrich referenced taxonomies while applying them to the construction phase use cases.

# 12 References and notes

1 Meta-Object Facility, https://en.wikipedia.org/wiki/Meta-Object_Facility

2 Saraju Mohanty, Chapter 12 Metamodel-Based Fast AMS-SoC Design Methodologies, "Nanoelectronic Mixed-Signal System Design", ISBN 978-0071825719 and 0071825711, 1st Edition, McGraw-Hill, 2015.

3 OMG Meta-Object Facility (MOF) Core Specification, Version 2.4.1". Object Management Group. p. 1. Retrieved 17 February 2014.

4 Ontology-based information modeling method for digital twin creation of as-fabricated machining parts, Sheng Daia, Gang Zhao, Yong Yu, Pai Zheng, Qiangwei Bao, Wei Wang

5 The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

6 Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

7 EXPRESS is a standard data modeling language for product data. EXPRESS is formalized in the ISO Standard for the Exchange of Product model STEP (ISO 10303), and standardized as ISO 10303-11

8 In computer science and information science, an ontology encompasses a representation, formal naming and definition of the categories, properties and relations between the concepts, data and entities that substantiate one, many, or all domains of discourse. More simply, an ontology is a way of showing the properties of a subject area and how they are related, by defining a set of concepts and categories that represent the subject.

9 https://www.w3.org/2007/OWL/wiki/MOF-Based_Metamodel

10 The Semantic Web is an extension of the World Wide Web through standards set by the World Wide Web Consortium (W3C). The goal of the Semantic Web is to make Internet data machine-readable.

11 In computing, linked data (often capitalized as Linked Data) is structured data which is interlinked with other data so it becomes more useful through semantic queries. It builds upon standard Web technologies such as HTTP, RDF and URIs, but rather than using them to serve web pages only for human readers, it extends them to share information in a way that can be read automatically by computers. Part of the vision of linked data is for the Internet to become a global database.

12 Generating Digital Twin models using Knowledge Graphs for Industrial Production Lines, Agniva Banerjee and al.

13 Hypertext Markup Language, Word Wide Web Consortium (W3C), https://fr.wikipedia.org/wiki/Hypertext_Markup_Language

14 Resource Description Framework (RDF), World Wide Web Consotium (W3C), https://www.w3.org/RDF/

15 A Holistic Digital Twin Based on Semantic Web Technologies to Accelerate Digitalization, Patrick Moder and al.

16 An Approach for Process Model Extraction by Multi-grained Text Classification, Dustdar and Falchuk 2006; Mane et al. 2019, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7266458/

17 THE WEB SEMANTIC A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, by TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA

18 The Fourth Industrial Revolution (4IR or Industry 4.0) is the ongoing automation of traditional manufacturing and industrial practices, using modern smart technology. Large-scale machine-to-machine communication (M2M) and the internet of things (IoT) are integrated for increased automation, improved communication and self-monitoring, and production of smart machines that can analyze and diagnose issues without the need for human intervention.

19 D. Avison, "MERISE: A European Methodology for Developing Information Systems", European Journal of Information Systems, Jan. 1991, p. 183-191

20 Mathias Bonduel, A Framework for a Linked Data-based Heritage BIM (dissertation-thesis), 2021, https://lirias.kuleuven.be/retrieve/618662

21 A Context-Aware Augmented Reality System to Assist the Maintenance Operators, E. Zhu, P. Ong and A.Y.C. Nee International Journal of Computer Integrated Manufacturing, 28 (2) (2015), pp. 213-225

22 Integrating Ontology Into PLM- Tools to Improve Sustainable Product Development R. Stark and A. Pfortner CIRP Annals - Manufacturing Technology, 64 (1) (2015), pp. 157-160

23 Semantic Data Management for the Development and Continuous Reconfiguration of Smart Products and Systems M. Abramovici, J. Gobel and H. Dang CIRP Annals - Manufacturing Technology, 65 (2016), pp. 185-188

24 Description Logics M. Krötzsch, S. Frantisek and I. Horrocks IEEE Intelligent Systems, 29 (1) (2015), pp. 12-19

25 The Internet of Things (IoT) describes physical objects (or groups of such objects), that are embedded with sensors, processing ability, software, and other technologies, and that connect and exchange data with other devices and systems over the Internet or other communications networks, Gillis, Alexander (2021). "What is internet of things (IoT)?". IOT Agenda. Retrieved 17 August 2021.

[26] In knowledge representation and reasoning, knowledge graph is a knowledge base that uses a graph-structured data model or topology to integrate data. Knowledge graphs are often used to store interlinked descriptions of entities – objects, events, situations or abstract concepts – with free-form semantics. Ehrlinger, Lisa; Wöß, Wolfram (2016). Towards a Definition of Knowledge Graphs. SEMANTiCS2016. Leipzig: Joint Proceedings of the Posters and Demos Track of 12th International Conference on Semantic Systems – SEMANTiCS 2016 and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16). pp. 13–16.

27 The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains: the nouns representing classes of objects and the verbs representing relations between the objects. Knublauch, Holger; Oberle, Daniel; Tetlow, Phil; Wallace, Evan (March 9, 2006). "A Semantic Web Primer for Object-Oriented Software Developers". W3C. Retrieved November 19, 2017.

28 "The MOUSE approach: Mapping Ontologies using UML for System Engineers". Chung, Seung-Hwa (2018). Computer Reviews Journal: 8–29. ISSN 2581-6640.

29 Tim Berners-Lee (2006-07-27). "Linked Data". Design Issues. W3C. Retrieved 2010-12-18.

30 SPARQL for SPARQL Protocol and RDF Query Language) is an RDF query language—that is, a semantic query language for databases—able to retrieve and manipulate data stored in Resource Description Framework (RDF) format. Hebeler, John; Fisher, Matthew; Blace, Ryan; Perez-Lopez, Andrew (2009). Semantic Web Programming. Indianapolis, Indiana: John Wiley & Sons. p. 406. ISBN 978-0-470-41801-7.

31 INSPIRE Directive https://inspire.ec.europa.eu/inspire-directive/2 , UML Class Diagram for Building Field that follows INSPIRES directive https://inspire.ec.europa.eu/data-model/approved/r4618-ir/html/

32 Mathias Bonduel, A Framework for a Linked Data-based Heritage BIM (dissertation-thesis), 2021, https://lirias.kuleuven.be/retrieve/618662

33 Ana Roxin. Vocabulaires de données liées pour le BIM. 5e édition EDUBIM (journées de l'enseignement et de la recherche sur la maquette numérique et le BIM en France), May 2019, Cachan, France. ffhal-02279638

34 BIMERR (H2020 project), https://bimerr.eu/

35 BIM4EEB (H2020 project), https://www.bim4eeb-project.eu/

36 Data Provenance Ontology, https://www.w3.org/TR/prov-o/

37 Key Performance Indicator Ontology, https://bimerr.iot.linkeddata.es/def/key-performance-indicator/

38 Quantities, Units and Dimension Ontology, http://qudt.org/

39 Simple Knowledge Organization System Ontology, https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html

40 The Industry Foundation Classes (IFC) data model is intended to describe architectural, building and construction industry data. "BIM". buildingSMART. Archived from the original on 2011-03-05. Retrieved 2017-03-19.

41 Construction Taxonomy (ConTax), https://mathib.github.io/ConTax-ontology/

42 Building Element Ontology, https://pi.pauwel.be/voc/buildingelement/index-en.html

43 Architecture Taxonomy, https://mathib.github.io/aat-arch-taxonomy/

44 Funriture Taxonomy, https://mathib.github.io/aat-furn-taxonomy/

45 Mechanical, Electrical and Plumbing Taxonomy, https://mathib.github.io/aat-mep-taxonomy/

46 Linking Geospatial Data With GeoSPARQL, Robert Battle, Dave Kolas, Knowledge Engineering Group, Raytheon BBN Technologies

47 Well-known Text Representation of Coordinate Reference Systems, Open Geospatial Consortium (OGC), https://www.ogc.org/standards/wkt-crs

48 GeoSPARQL - A Geographic Query Language for RDF Data, Open Geospatial Consortium (OGC), https://www.ogc.org/standards/geosparql

49 Geography Markup Language, Open Geospatial Consortium (OGC), https://www.ogc.org/standards/gml

50 BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data, Chi Zand and Al., https://www.researchgate.net/publication/326965970_BimSPARQL_Domain-specific_functional_SPARQL_extensions_for_querying_RDF_building_data

51 A Framework for a Linked Data-based Heritage BIM, Mathias Bonduel

52 Ontology for Managing Geometry (OMG), https://www.projekt-scope.de/ontologies/omg/

53 File Ontology for Geometry Format (FOG), https://mathib.github.io/fog-ontology/

54 Geometry Metadara ontology (GOM), https://mathib.github.io/gom-ontology/

55 A. Borrmann, M. König, C. Koch, and J. Beetz. “Building Information Modeling: Why? What? How?” In: Building Information Modeling. Ed. By A. Borrmann, M. König, C. Koch, and J. Beetz. Springer International Publishing, 2018. Chap. 1, pp. 1–24. isbn: 978-3-319-92861-6. doi: 10. 1007/978-3-319-92862-3_1.

56 P. Pauwels, S. Zhang, and Y.-c. Lee. “Semantic web technologies in AEC industry: A literature overview”. In: Automation in Construction 73 (2017), pp. 145–165. doi: 10.1016/j.autcon.2016.10.003.

57 EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology; Pauwels, P. and Terkaj, W; Automation in Construction 63 (2016) 100-133.

58 Building Topology Ontology, https://w3c-lbd-cg.github.io/bot/

59 BOT: the Building Topology Ontology of the W3C Linked Building Data Group; Rasmussen, M. H., Lefrançois, M., Schneider, G. F., and Pauwels, Pieter; Semantic Web (Nov 2020). DOI:10.3233/SW-200385.

60 Tarcisio Mendes de Farias, Ana Roxin, Christophe Nicolle. COBieOWL, an OWL ontology based on COBie standard. In proceedings of On the Move to Meaningful Internet Systems: OTM 2015 Conferences. Lecture Notes in Computer Science 9415, Springer 2015.

61 Building Topology Ontology, https://w3c-lbd-cg.github.io/bot/

62 This group brings together experts in the area of building information modelling (BIM) and Web of Data technologies to define existing and future use cases and requirements for linked data based applications across the life cycle of buildings. https://www.w3.org/community/lbd/

63 BIM and ISO 19650 from a project management perspective, EFCA BIM Task Force, ISBN 9789075085075

64 Time Ontology, https://w3c.github.io/sdw/time/

65 Ontology-Based Information Integration and Decision Making in Prefabricated Construction Component Supply, Juan Du and al., Chain https://core.ac.uk/download/pdf/301372081.pdf

66 A Security Ontology for Security Requirements Elicitation, Amina Souag and al., https://hal-paris1.archives-ouvertes.fr/hal-01153319/document

67 Damage Topology Ontology, Al-Hakam Hamdan and al., https://alhakam.github.io/dot/

68 Ontology-Based Information Integration and Decision Making in Prefabricated Construction Component Supply, Juan Du and al., Chain https://core.ac.uk/download/pdf/301372081.pdf

69 BPMN Based Ontology (BBO), https://www.irit.fr/recherches/MELODI/ontologies/BBO/index-en.html

70 ISO/IEC 19510:2013 Information technology — Object Management Group Business Process Model and Notation

71 BUSINESS PROCESS MODEL AND NOTATION SPECIFICATION VERSION 2.0.2, https://www.omg.org/spec/BPMN/2.0.2/

72 Construction Task Ontology, Mathias Bonduel, https://mathib.github.io/cto-ontology/

73 Digital Construction Processes Ontology, Seppo Törmä, https://digitalconstruction.github.io/Processes/v/0.3/

74 Digital Construction Site Twin Dictionnary , CSTB, https://cesium.cstb.fr/Apps/bim2twin/dico_bim2twin.htm

75 Huhnt, W. (2009). Process modelling in civil engineering. Journal of the International Association for Bridge and Structural Engineering, 19 (1), 91–101. https://doi.org/10.2749/101686609787398317

76 A Uniform Resource Identifier (URI) is a unique sequence of characters that identifies a logical or physical resource used by web technologies. URIs may be used to identify anything, including real-world objects, such as people and places, concepts, or information resources such as web pages and books.

77 Well-known Text Representation of Coordinate Reference Systems, Open Geospatial Consortium (OGC), https://www.ogc.org/standards/wkt-crs

78 Named graphs are a key concept of Semantic Web architecture in which a set of Resource Description Framework statements (a graph) are identified using a URI, allowing descriptions to be made of that set of statements such as context, provenance information or other such metadata.