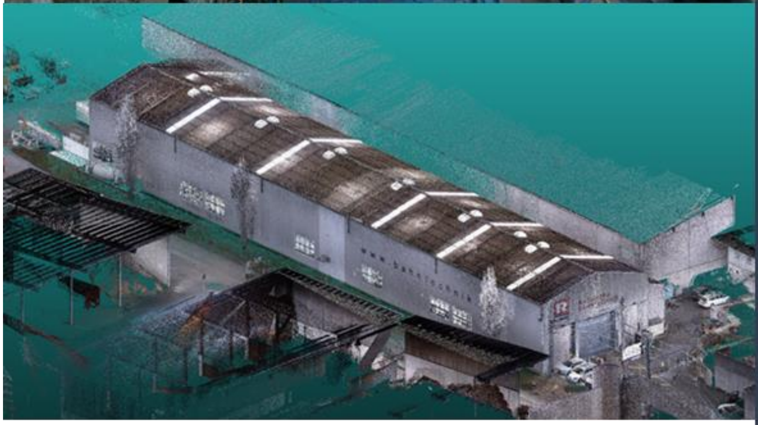


COGITO

CONSTRUCTION PHASE
DIGITAL TWIN MODEL

cogito-project.eu

D3.1 Survey of Existing Models, Ontologies and Associated Standardization Efforts

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310

D3.1 – Survey of Existing Models, Ontologies and Associated Standardization Efforts

Dissemination Level: Public

Deliverable Type: Report

Lead Partner: UPM

Contributing Partners: Hypertech, UCL, AU, UEDIN, CERTH, BOC, QUE, NT, FER

Due date: 30-04-2021

Actual submission date: 29-04-2021

Authors

Name	Beneficiary	Email
Socorro Bernardos	UPM	sbernardos@fi.upm.es
Alba Fernández-Izquierdo	UPM	albafernandez@fi.upm.es
Raúl García-Castro	UPM	rgarcia@fi.upm.es
Panos Andriopoulos	QUE	mailto:panos@que-tech.com
Vladislav Baňas	NT	banas@novitechgroup.sk
Frédéric Bosché	UEDIN	f.bosche@ed.ac.uk
Martin Bueno Esposito	UEDIN	martin.bueno@ed.ac.uk
Michalis Chatzakis	CERTH	mchatzak@iti.gr
Christos Chronopoulos	AU	chrichr@cae.au.dk
Chris Dent	UEDIN	chris.dent@ed.ac.uk
Lukas Esterle	AU	Lukas.esterle@ece.au.dk
Damiano Falcioni	BOC	damiano.falcioni@boc-eu.com
Jozef Fedor	NT	fedor@novitechgroup.sk
Giorgos Giannakis	Hypertech	g.giannakis@hypertech.gr
Apostolia Gounaridou	CERTH	agounaridou@iti.gr
Tobias Hanel	FER	thanel@ferrovial.com
Vasilios Karkanis	CERTH	vkarkanis@iti.gr
Panagiotis Moraitis	QUE	p.moraitis@que-tech.com
Gail Robertson	UEDIN	gail.robertson@ed.ac.uk
Tasos Sinanis	CERTH	atsakir@iti.gr
Martin Straka	NT	straka@novitechgroup.sk
Jochen Teizer	AU	teizer@eng.au.dk
Katerina Valalaki	Hypertech	k.valalaki@hypertech.gr
Ján Varga	NT	varga@novitechgroup.sk
Michalis Vasileiadis	Hypertech	m.vassiliadis@hypertech.gr
Amy Wilson	UEDIN	Amy.l.wilson@ede.ac.uk
Georgios Lilis	UCL	g.lilis@ucl.ac.uk
Kyriakos Katsigarakis	UCL	k.katsigarakis@ucl.ac.uk
Robert Woitsch	BOC	robert.woitsch@boc-eu.com

Reviewers

Name	Beneficiary	Email
Alexandros Archontidis	OLOD	aarchontidis@olympiaodos.gr
Elias Bruno Meusburger	RSRG	Elias.Meusburger@rsrg.com

Version History

Version	Editors	Date	Comment
0.1	Raúl García-Castro, Socorro Bernardos	22-01-2021	First draft including an example for partners to know what is expected from them
0.2	Juan Cano-Benito, Socorro Bernardos	02-02-2021	EthOn added
0.3	NT	05-05-2021	I3D added
0.4	Socorro Bernardos	15-02-2021	SAREF4BLDG, W3C BOT, ifcOWL added
0.5	Alba Fernández-Izquierdo	16-02-2021	CityGML added
0.6	Martin Bueno, Socorro Bernardos	03-03-2021	Contributions by UEDIN, references updated, Figure captions as template
0.7	Socorro Bernardos	10-03-2021	Owl-Time, saref4ehaw, saref4wear added
0.8	NT	11-03-2021	More on i3d data model
0.9	UEDIN (Gail), Socorro Bernardos	17-03-2021	Statistical modelling description and STATO added; some sections revised
0.10	UCL, Socorro Bernardos	18-03-2021	Building description, IFC, Brick Schema and Haystack model added, some sections revised
0.11	CERTH, Socorro Bernardos	19-03-2021	Chapters 2.3.1, 4.1.1.2, 7.1.1.3 and 7.1.2.3 added; some formatting done
0.12	Socorro Bernardos	22-03-2021	BPO added
0.13	Alba Fernández-Izquierdo	23-03-2021	Introduction and executive summary added. Added more information about ifcOWL.
0.14	Hypertech, Socorro Bernardos	23-03-2021	Sections 2.3.2, 4.2.1, 4.2.2.4 added, some editing done
0.15	QUE, Socorro Bernardos	23-03-2021	Sections 5.1.1.4, 5.2.2.2 added; some editing done
0.16	Tobias Hanel (FER), Socorro Bernardos	23-03-2021	Section 2.2.2 added; some editing done
0.17	Socorro Bernardos	25-03-2021	Section 2.1 added, several sections revised and commented on
0.18	CERTH, Socorro Bernardos	26-03-2021	Chapters 2.3.1, 4.1.1.2, 7.1.1.3 and 7.1.2.3: Changes and additions based on comments.
			Table in conclusions
0.19	Gail Robertson, Socorro Bernardos	30-03-2021	More on statistical modelling: references, STAT; acceptance of deletion of some texts
0.20	Socorro Bernardos	30-03-2021	Abbreviations added
0.21	CERTH	31-03-2021	Suggestion to move an ontology from 7.1.2 to 4.1.2
0.22	UCL	02-04-2021	Section3.2.1.1 added
	BOC		Sections 2.2.4, 3.4.1.2, 5.1.1.1 added
	Socorro Bernardos		Input revised
0.23	Alba Fernández-Izquierdo	07-04-2021	Section 3.4.2.2 added. Changes and additions added based on comments.
	Panagiotis Moraitis		References added
	Raúl García-Castro		Comments added

	Socorro Bernardos		Integration of contributios and some responses to comments
0.24	Socorro Bernardos	08-04-2021	DICO ontologies added
0.25	Socorro Bernardos	12-04-2021	Section 2 revised: new organisation
0.26	BOC	13-04-2021	Changes in sections 4.1.2 and 6.1.1.1 according to comments
	Socorro Bernardos		Section 3 re-organised to match section 2
0.27	Alba Fernández Izquierdo	13-04-2021	Conclusions added. Executive summary updated.
	Socorro Bernardos	13-04-2021	Changes to section 1.3, added relevance of several resources to COGITO
0.28	Raúl García-Castro	14-04-2021	Minor changes across the document
0.29	Socorro Bernardos	19-04-2021	Section 7 revised from new input by AU and comments by UEDIN
			Sections 3 and 4 revised according to review by QUE
			Some references for section 2 added
0.30	Socorro Bernardos	23-04-2021	BIMMER ontologies added, some comments on CityGML
	Alba Fernández-Izquierdo		
1.0	Socorro Bernardos Galindo	27-04-2021	Changes according to reviews by OLOD and RSRG

Disclaimer

©COGITO Consortium Partners. All right reserved. COGITO is a HORIZON2020 Project supported by the European Commission under Grant Agreement No. 958310. The document is proprietary of the COGITO consortium members. No copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights. The information in this document is subject to change without notice. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use, which may be made, of the information contained therein.

Executive Summary

This deliverable presents the results of a survey on existing data models and ontologies that are relevant to the COGITO information domains' requirements. The survey has been conducted in the context of task "T2.1 - Survey of Existing Data Models & Ontologies & Associated Standardization Efforts". The data models and ontologies documented herein are related to several domains relevant to COGITO that the consortium has identified at this stage of the project, i.e., building, process, multi-source visual data, Internet of Things (IoT), simulation, workflow management and smart contracts, health and safety, and quality control. Twelve data models and twenty-three ontologies have been extracted from the literature and briefly presented in this document.

The aim of this document is to provide an overview of existing data models and ontologies to be considered for reuse and alignment during the development of the COGITO ontology. Moreover, this survey also aims at identifying potentially missing elements to cover the information scope in COGITO.

From the results of the survey, it has been observed that, in general, there are exiting ontologies and data models related to the majority of the identified domains, with the exception of the multi-source visual data and workflow management domains, where no specific ontologies tailored to their scope were found. The survey has also revealed that well-known ontologies related to the Internet of Things, simulation and construction crew and safety domains could be considered as potential ontologies for reuse within COGITO. Two ontologies and two data models appear to be cross-domain and that are relevant to the majority of the COGITO domains presented in this document, i.e., the IFC and cityGML data models and the DICO and ifcOWL ontologies.

This document will constitute the main input in task "T3.2 – COGITO Data Model, Ontology Definition and Interoperability Design", where the COGITO consortium will determine to what extent these identified ontologies and data models can be reused for the development of the COGITO ontology.

Table of contents

List of Figures

List of Tables

List of Abbreviations, Acronyms and Namespaces

Term	Description
AEC	Architecture Engineering and Construction
AECOO	Architecture, Engineering, Construction, Ownership and Operation
AHU	Air Handling Unit
AR	Augmented Reality
ASTM	American Society for Testing and Materials
BAS	Building Automation System
BBO	BPMN Based Ontology
BCF	BIM Collaboration Format
BIM	Building Information Modelling / Building Information Management
BMS	Building Management System
BOT	Building Topology Ontology
BPMN	Business Process Modelling Notation
BPO	Building Product Ontology
bSI	buildingSMART International
CDE	Common Data Environments
COGITO	Construction Phase diGItal Twin mOdel
DL	Description Logic
DT	Digital Twin
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
FM	Facilities Management
FOAF	Friend Of A Friend
GIS	Geographic Information System
H&S	Health and Safety
HVAC	Heating Ventilation and Air Conditioning
ICT	Information and Communication Technology
IFC	Industry Foundation Classes
IoT	Internet of Things
IOC	IoT Oriented Criterion
ISO	International Standard Organisation
KPI	Key Performance Indicator
LOV	Linked Open Vocabularies
MEP	Mechanical, electrical and plumbing
OASIS	Ontology for Agents, Systems, and Integration of Services
ODM	Observations Data Model
OGC	Open Geospatial Consortium
OWL DL	Web Ontology Language Description Logic
PDCA	Plan-Do-Check-Adjust
PLM	Product Lifecycle Management
PROV-O	Provenance Ontology
QA	Quality Assurance

QC	Quality Control
QMS	Quality Management System
QUDT	Quantities, Units, Dimensions and Types
RDF	Resource Description Framework
SAREF	Smart Applications REFerence ontology
SKOS	Simple Knowledge Organisation System
SSN	Semantic Sensor Network
SOSA	Sensor, Observation, Sample and Actuator
SPARQL	SPARQL Protocol and RDF Query Language
TD	Thing Description
W3C	World Wide Web Consortium
WOC	Web Oriented Criterion
WoT	Web of Things
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Prefix	Namespace
bot	https://w3c-lbd-cg.github.io/lbd/bot/
bpo	http://www.w3id.org/bpo#
geo	http://www.w3.org/2003/01/geo/wgs84_pos#
s4bldg	https://saref.etsi.org/saref4bldg/
s4city	https://saref.etsi.org/saref4city/v1.1.2/
s4ehaw	https://saref.etsi.org/saref4ehaw/v1.1.1
s4wear	https://saref.etsi.org/saref4wear/v1.1.1/
saref	https://saref.etsi.org/core/
sosa	http://www.w3.org/ns/sosa/
ssn	http://www.w3.org/ns/ssn/
ssn-system	http://www.w3.org/ns/ssn/systems/
time	https://www.w3.org/TR/owl-time/
xsd	http://www.w3.org/2001/XMLSchema#

1 Introduction

1.1 Scope and Objectives of the Deliverable

Building Information Modelling (BIM) [1] and reality capture technologies in the construction phase have gained significant attention as of late, revealing the need for an efficient usage and exchange of data across the various application areas that the construction projects entail. In practice, large scale construction projects often require collaboration and information exchange among different actors and systems. Within COGITO, a multidimensional, dynamically updated, digital representation (twin) of a construction project is introduced, using methods to ensure interoperability among the different components and technologies.

The Digital Twin platform that will constitute the backbone of the COGITO solution aims to enable interoperability with existing and emerging standards and data formats covering various domains. Interoperability could be applied to different levels in information systems, namely, technical interoperability, syntactic interoperability and semantic interoperability. Technical interoperability addresses the problems related to the hardware and software elements involved in the communications protocol, infrastructures, and physical components to exchange data. Syntactic interoperability focuses on the exchange of data formats between systems. Finally, semantic interoperability aims at providing successful exchange and processing of meaningful information in addition to raw data.

This deliverable presents a survey of the existing data models, ontologies, standards and proprietary data models of prominent ICT tools that can be used in the COGITO environment for exchanging data. The goal of this document is to analyse the literature to identify what already exists and is available in the domains that are relevant to the scope of COGITO. Considering the fact that additional domains might arise during the project lifetime, at this stage of the project, the domains that are anticipated to be studied in terms of the semantic modelling are the building, processes, multi-source visual data, Internet of Things (IoT), simulation, workflow management and smart contracts, health and safety, and quality control domains.

1.2 Relation to other Tasks and Deliverables

This deliverable is based on the use cases defined in D2.1 and T2.1 *Stakeholder requirements for the COGITO system*; and it will be used to identify potentially missing elements to cover the information scope of COGITO and develop the COGITO ontologies (task T3.2) to support semantic interoperability in the project.

1.3 Structure of the Deliverable

The deliverable is structured as follows:

- **Section 2** provides an overview of the domains that have been identified as relevant to the COGITO activities at this stage of the project;
- **Section 3** presents data models, ontologies and standards that are related to the construction site domain, focusing on buildings;
- **Section 4** presents data models, ontologies and standards related to the construction process;
- **Section 5** showcases data models, ontologies, and standards that could partially but semantically represent data that are captured by the as-built information via reality capture tools (multi-source visual data, IoT and simulation);
- **Section 6** introduces data models, ontologies, and standards which are relevant to the workflow management and smart contracts domains;
- **Section 7** summarises data models, ontologies and standards related to the health and safety domain;
- **Section 8** summarises the relevant data models, ontologies and standards related to quality control;
- and finally, **Section 9** concludes this work and provides some insights into the future use of this survey results.

2 Relevant Information Domains

2.1 Digital Twins in the Construction Phase

During the last two decades, the Architecture Engineering and Construction (AEC) industry is evolving with the incorporation of various computer technologies enabling modelling of whole construction site or building assets along with their components (structural and non-structural), providing the involved AEC stakeholders with a shared resource of information which can form a reliable basis for decisions and actions throughout a project's lifecycle. The traditional paper-based information (e.g. drawings, specifications, etc.) collection and management is currently shifting to electronic documents, file-based drawings, and digital data recording.

Adoption of emerging technologies such as Building Information Modelling (BIM) is triggering a digital transformation of the global AEC industry. Building Information Modelling (also known as Building Information Managing) is based on managing Information Models, which digitally represent the asset (building or infrastructure) to be projected, constructed or operated [2]. An Information Model is a set of structured and non-structured information containers, being BIM Models the main structured container, so the rest of information inside the other containers should reference to it (ISO 19650-1:2018). In a BIM model, the representation of an element is defined through its physical and functional characteristics using parameters and properties [3]. This information is specified progressively during the asset's life cycle, from design to operation. Moreover, different stakeholders simultaneously develop the BIM Model, storing on it the product of their decisions. To produce this collaborative works it has to be set procedures, ICT tools architectures and formats (e.g. IFC or BIM Collaboration Format (BCF)) inside what is commonly named Common Data Environments (CDE) [4].

All this information is subject to the traceability of the changes that are produced during the entire life cycle of the asset. Usually, one system is used throughout the Delivery phase (Design and Constructing) and another, during the Operation period [5].

According to [6], the Digital Twin idea was first introduced as an unnamed concept for Product Lifecycle Management (PLM) back in 2002 and was subsequently called Mirrored Spaces Model, Information Mirroring Model and even Virtual Twin until its final denomination as Digital Twin in 2011. Since this, the term has been widely used in different sectors for referring to digital replicas of physical entities.

The AEC sector has not been alien to the idea of Digital Twins, and as part of the process of technological change happening in the last few years, more and more interest has arisen in the concept of Digital Twin [6].

We could say that a Building Information Model provides the basis for a Digital Twin, yet it does not provide direct physical-digital linkages to link information about the current behaviour or state and as such does not serve as a virtual operation tool [7]. There is a lot of information from the building and its subsystems which extend current BIM methodology including the integration with IoT (Internet of Things), BMS (Building Management System), ERP (Enterprise Resource Planning) or BAS (building automation system), to provide tools to capture, store, and share critical building information [8].

In COGITO, the Digital Twin platform will handle and store all construction data to bring all the as-designed and as-built information together to create a digital representation of construction projects enabling computer-aided analysis and decision making via construction services. To support these goals, the construction site, processes and management need to be properly modelled based on a common information model, while data must be captured to populate that model. Especial attention will be paid to assist the delivery of services to safety assurance, quality control, and workflow modelling and management. Therefore, the following domains have been chosen to be analysed in this report:

- Construction site:
    - Building;
- Construction process;
- Reality-capture:
    - Multi-source visual data

    - Internet of Things
    - Simulation
- Applications
    - Workflow management and smart contracts
    - Construction safety
    - Quality control

2.2 Construction Site

Diverse data models and ontologies derived from existing standards could be potential data formats to be linked towards developing a multidimensional BIM-based ontology for building and infrastructure construction sites that meet the COGITO interoperability requirements. In contrast to the Buildings domain, where various data models and ontologies exist to semantically link multilevel information, the Infrastructures domains has so far been overlooked. Although there is an upcoming standardised data mode, the Industry Foundation Class – IFC4.3 release candidate[1], that support many infrastructure types (e.g., railways and roads), it has not been widely tested. Thus, we decided to exclude the infrastructure domain from this study, and further investigate it at the early stages of "T3.2 – COGITO Data Model, Ontology Definition and Interoperability Design" activities. It is worth mentioning though that the aforementioned data model inherits concepts from the IFC4 that is briefly presented in section 3.1.1.1.

2.2.1 Building

A building is a set of components related to the architectural, structural, and building systems domains of Architectural Engineering and Construction sector. These elements are grouped into nested subsets referring to the stories, apartments and internal spaces and opening volumes following a tree-like structure. As far as construction management is concerned, in COGITO the building model elements are grouped with respect to respective time intervals related to different construction phases.

The architectural building elements refer to components of the building fabric which play a more aesthetic role to the overall building structure (e.g. slabs, walls, opening volumes) as opposed to the structural building elements which are vital to the support of the overall building structure (e.g. beams, columns). Building systems refer to devices which both consume energy and contribute to the building operation.

Additionally, many of the building components are also annotated with several properties. Some of these components appear as elements in different product catalogues, referring to existing products in the market. For example, a building wall is connected to a specific material layer bedding, where each material has specific thermal properties, or a building boiler is connected to a set of performance measure values.

2.3 Construction Process

"Gartner defines business process as an event-driven, end-to-end processing path that starts with a customer request and ends with a result for the customer. Business processes often cross departmental and even organizational boundaries.[2]" Since the 1990s[3] the business process is understood as a sequence of actions that are manually, semi-automatically or automatically performed aiming to achieve the company's goals.

This leads to following conclusions: (a) The business processes are aligned to company's / customers goals and (b) The business processes are executed on different – social, virtual, physical - platforms.

The construction process is not just how something is built; it is all the steps involved around the building of something. From site choice to final inspection (and all the thousands of conversations with contractors and

[1] https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC1/HTML/

[2] https://www.gartner.com/en/information-technology/glossary/business-process

[3] Michael Hammer, James Champy: *Business Reengineering. Die Radikalkur für das Unternehmen.* 5. Auflage. Campus-Verlag, Frankfurt/ New York 1995, ISBN 3-593-35017-3

stakeholders in between), the construction process covers far more than the actual construction phase itself. It can be broken down into 6 main parts[4]:

- Conception;
- Design;
- Pre-construction;
- Procurement;
- Construction;
- Post-construction.

The construction part is the physical processes of building, landscaping or refurbishing plus all the associated activities, such as demolition, site clearance, administration and so on.

In COGITO, the different phases of a construction project need to be represented in order to be able to analyse the tasks that have taken place and suggest the following procedures.

2.4 Reality Capture

Within COGITO, reality capture tools will collect all the necessary data regarding the status and conditions of the construction site via imaging, laser scanning, localisation, and other means.

Data will flow continuously from these tools to the platform, reflecting the actual state and progress/evolution of the construction site. Raw data captured by reality capture tools will serve as input to the pre-processing modules (Visual and IoT), which will populate data requested for the COGITO construction Digital Twin Applications (Visual & Geometric Quality Control, Health & Safety, and Workflow Simulation & Management).

2.4.1 Multi-source Visual Data

Devices that might be used for capturing the visual data could be any device that includes a built-in camera, for example a smartphone, a tablet, a head mounted display device (such as AR goggles), a satellite, a video camera, a, infrared or thermal cameras, etc. Those images might be direct captures as an image file or might be a frame capture of video. They might also be infrared images. Infrared images can be captured from infrared cameras and basically expand the portion of the spectrum that the human eye can see, by visualizing and measuring the thermal energy that is emitted from an object. Everything with an emitting temperature above absolute zero (0 degrees in Kelvin) emits heat that can be displayed in an infrared image.

There are two basic methods used for the description of images and image contents. The image data can be treated as a physical image representation and their meaning as a logical image representation [9]. The most common form of the physical image representation is the raster form. It includes the image header (describes the main image parameters, such as formal, resolution, bits per pixel number and compression information) and the image matrix (contains the image data). In the logical image representation, the description of the image object includes meta, semantic, colour, texture, shape and spatial attributes. Many of the features can be represented as sets of points tagged with labels to capture necessary semantics.

2.4.2 Internet of Things

Apart from the multi-source visual data, an Internet of Things (IoT) solution that (a) feeds the IoT Data Pre-processing Module with raw IoT streams from the construction field, and (b) meet the IoT data requirements of the COGITO Digital Twins (DT) Applications, will be designed, developed, and delivered. More specifically, real-time data acquisition about the crew, heavy machinery and other resources' location tracking will be leveraged by a subset of Digital Twin Applications, such as the Proactive Health and Safety and the Workflow Monitoring. Initially, a thorough survey on a wide range of hardware IoT technologies will be performed, followed by an evaluation of available communication protocols. Among many, such as computer vision, ultrasonic and infrared, the survey will emphasize to "non-line of sight" methods, with the Radio Frequency Identification (RFID)-based

[4] https://monday.com/blog/construction/what-is-a-construction-process/

the Global Navigation Satellite System (GNSS)-based and Bluetooth being the most promising. The aforementioned survey will conclude to the IoT sensor network that the COGITO project will propose as the building and infrastructure construction phase IoT solution.

Regardless the results of the survey, the IoT solution will encompass numerous IoT devices whose data formats and APIs will be heterogeneous. Devices from different vendors, which will be considered, may represent data in different formats, and even when a common format is used, the internal data model schema may vary. Such a diversity derives in semantic interoperability obstacles since each system can represent the same thing in different ways. Apparently, the semantic modelling of heterogeneous devices is a non-trivial task. The development of an IoT model tailored to project's requirements is enticing, however it entails the risk of creating a model which is not extensible beyond the initial application and reduces interoperability with other existing solutions.

Semantic Technologies, and more precisely ontology-based approaches, can be leveraged to remedy the aforementioned drawbacks. Among a wide range of available ontologies for IoT device metadata modelling and semantics annotation, ontologies that are supported by standardisation bodies are introduced in Section 5.2.1

2.4.3 Simulation

The aim of COGITO is to combine as-planned data with real-world data so that the simulation models can be calibrated to the actual state on construction sites. As much of the data is location data or time-series data, methods that account for spatial-temporal correlations (e.g. ARIMA modelling, kriging) can be very useful. Datasets created from real-time construction site data will be noisy and careful organisation of datasets will be needed to combine different types of data to be used in validation of simulation models. Information regarding proximity of workers and equipment in space and time and equipment sensor data will be collated from construction site data to infer worker behaviour and equipment use on site. Network analysis methods and graphical modelling approaches (e.g. Bayesian networks [10], Chain Event Graphs [11]) will be investigated for capturing the dependencies exhibited by the datasets associated with different activities.

Once the simulation model has been developed and calibrated it will be used to optimise processes and workflow. Some of the inputs to the simulation model are uncertain (e.g. unexpected weather, inaccurate sign-in records) – if workflows are to be optimised robustly against the real-world (rather than the model) this uncertainty must be modelled and accounted for. Some uncertainties arise from extreme events (e.g. weather) so methods that deal with rare events (e.g. extreme value theory) will be tested. Other uncertainties may not have any associated data so methods for elicitation of probabilistic judgments from experts will be used. Bayesian statistical emulators will be explored for speeding up simulations.

In this regard, data models and ontologies related to the simulation domain can be relevant to COGITO.

2.5 Applications

2.5.1 Workflow Management and Smart Contracts

Workflow management is an integral part of a construction industry processes. It is used to control activities needed from the construction's idea conception, through architectural and construction activities to the successful handover or even construction operation, depending on customer's needs and contractual obligations agreed. Formalisation of the workflows in the real world highly depends on project complexity. Small private projects usually rely on informal control and contractor experience. More complex projects incorporate workflow control through the construction schedule and experience of the project manager, who is responsible for managing all anticipated and planned work. To follow and facilitate these real-world activities, workflow management tools are to be incorporated into the digital twin counterpart created in the COGITO. It will lead to better control, automatization, and optimisation of all processes throughout the project lifecycle.

Nowadays blockchain has become a pervasive technology in a wide range of sectors such as finance, security, IoT or public services. A blockchain is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block so that once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks. One key emerging use case of blockchain technology involves “smart contracts”. Smart contracts are basically computer programs that can automatically execute the terms of a contract. When a pre-configured condition in a smart contract among participating entities is met then the parties involved in a contractual agreement can be automatically made payments as per the contract in a transparent manner [12]. In COGITO, smart contracts will be introduced to the workflow modelling and automation process to enhance transparency and will provide trusted means to verify completion of construction tasks, asset release, etc. to facilitate automated financial exchanges and minimise transaction costs and administrative overhead in construction.

2.5.2 Construction Safety

Fatal accidents in construction remain on the top of occupational accident types’ list in the European Union [13] and therefore, construction health and safety planning and management is an essential part of construction projects. Construction health and safety aims to establish a safe and healthy occupational environment for the construction crew as well as to foster a safety culture in the industry. According to the definition of occupational health by the World Health Organisation (WHO), occupational health and safety focuses strongly on hazard prevention. To achieve that, several rules and regulations exist in national level, which have been supplemented by the EU guidelines for the construction industry, such as the EU Directive 92/57/EEC, which outlines the minimum mandatory safety and health provisions for any construction site and sets the obligation to develop a Safety and Health Plan. In construction, safety planning typically includes the risk and hazard identification process and the selection of corresponding safety measures [14]. International standards, such as ISO 31000 set the principles and guidelines for risk identification, analysis and evaluation.

COGITO aims to integrate modern information and communication technologies to enhance health and safety in construction. Automated safety rule checking on continuously changing BIM data will allow for proactive occupational safety on construction sites. Additionally, artificial intelligence and machine learning for human and machinery trajectory prediction will potentially enable us to foresee hazardous situations and hence, improve health and safety in construction.

2.5.3 Quality Control

Quality Control (QC) in a construction project is a critical step to guarantee that the built elements meet the quality specifications (e.g. geometric tolerances) in order to ensure that the project “achieves the intended level of safety and serviceability” [15]. QC is also critical to ensure that the subsequent construction works are not delayed, mainly due to unplanned remedy reworks, resulting in additional costs.

QC complements (or may also be considered part of) *Quality Assurance (QA)* that setups a quality management system (QMS) that ensures that processes (and/or products) meet specifications and regulatory requirements. A QMS is typically built on the Plan-Do-Check-Adjust (PDCA) cycle method to ensure continuous improvement. PDCA is promoted in ISO 9001 [16], the benchmark international standard on Quality Management Systems.

The current practice in Geometric and Visual QC relies heavily on human input, which is prone to error, inefficient, and poorly connected with digitalisation efforts. For example, while geometric surveying practice is gradually (albeit slowly) adopting modern surveying technologies like Laser Scanning, the analysis of the acquired data to ensure that regulatory and project-specific specifications (tolerances) are met remains a manual and tedious process, prone to error.

COGITO aims to transform current Geometric and Visual QC processes by developing tools that process 3D point clouds and 2D images from sites to automatically detect defects based on industry standard and organisation specifications and record them to the Digital Twin platform for enhanced quality information retrieval and visualisation. This shall result in significant performance gains, in terms of productivity and fewer occurrences of missed defects.

Taking the above example of Geometric QC, COGITO will analyse Laser Scanning data automatically to ensure regulatory and project-specific specifications (tolerances) are controlled systematically and very efficiently.

Besides, integration with the DT will ensure that the geometric QC results are easily retrieved and visualised for fast information retrieval and decision making.

3 Construction Site

3.1 General

3.1.1 Relevant Data Models

3.1.1.1 IFC

The *Industry Foundation Classes* (IFC) is a standardised schema for describing building and non-building related model objects. IFC represents an open specification for BIM data that is exchanged and shared among various stakeholders engaged in a building/infrastructure construction or facility management project. Each IFC release is a candidate international openBIM standards, published by building Smart International[5] (bSI) (the latest standardised version appears in ISO 16739, 2018).

Its data are physically stored in a STEP file (readable with a text editor) using the EXPRESS data definition language. In addition to the IFC-EXPRESS specification an ifcXML specification is published as well, following the XML document structure.

The IFC specification includes terms, concepts and data specification items that originate from their use within disciplines, trades, and professions of the construction and facility management industry sector. The data schema architecture of IFC defines four conceptual layers, while an individual schema is assigned to each conceptual layer. Figure 1 summarizes this layered schema IFC architecture.

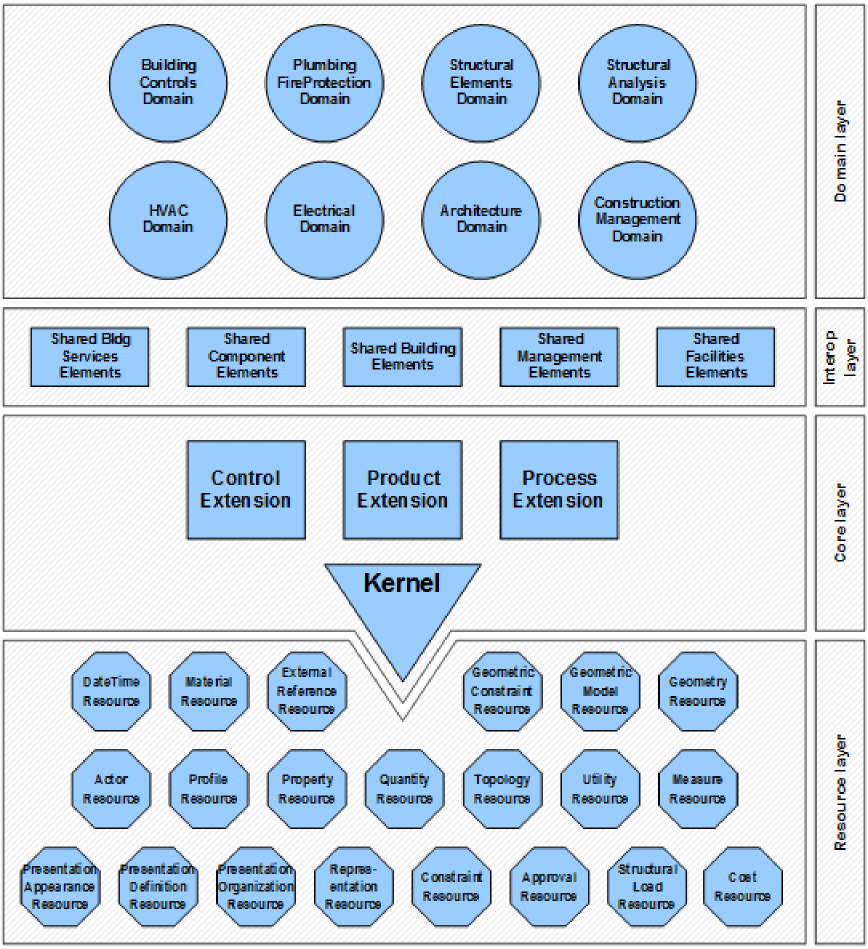


Figure 1 – IFC layered architecture[6]

[5] https://www.buildingsmart.org/

[6] https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/introduction.htm

The four layers of IFC schema are the following:

1. **Resource layer** — the lowest layer includes all individual schemas containing resource definitions; those definitions do not include a globally unique identifier and shall not be used independently of a definition declared at a higher layer.
2. **Core layer** — the second to lowest layer includes the kernel schema and the core extension schemas, containing the most general entity definitions, all entities defined at the core layer, or above carry a globally unique id and optionally owner and history information.
3. **Interoperability layer** — the second to highest layer includes schemas containing entity definitions that are specific to a general product, process or resource specialization used across several disciplines, those definitions are typically utilized for inter-domain exchange and sharing of construction information.
4. **Domain layer** — the highest layer includes schemas containing entity definitions that are specializations of products, processes or resources specific to a certain discipline, those definitions are typically utilized for intra-domain exchange and sharing of information.

IFC has a very accurate and detailed structure supporting various geometric representations for building and site elements which can be adopted for all COGITO geometry related operations. In addition, data from the Architectural, Structural, Construction management, Electrical, Plumping and HVAC domains could also be used in the project.

3.1.1.2 CityGML

CityGML is an open data model[7] which has been recognised as a standard of the OGC since 2008. CityGML models have been developed to reach a common definition of the basic entities, attributes and relations of a 3Dmodel that goes beyond the building level and aims to represent cities, districts or landscape. They have been widely used as they provide features to share and manage the complexity of a city. The use of CityGML for the 3D digital twinning of construction projects is an option up for consideration.

The CityGML standard contains definitions of the most important objects and their categories for the city information model. The CityGML information model consists of two types of modules: the core module and the extension modules. The core module contains the basic concepts and components of the CityGML information model. Extension modules ensure that the addition of new thematic classes to the city information model is possible. The most important objects of the city information model are divided into thematic modules according to their characteristics. These CityGML modelled themes include terrain models, buildings, bridges, water and traffic areas, vegetation and street furniture, etc. Appearance can be added to three-dimensional city objects, and this can be any form of visualization. The appearance can visualize items according to the shape, texture, materials or themes. The Application Domain Extension (ADE) of the CityGML information model is an extension feature that allows new items or properties to be added to the information model. The ADE can be also used to add energy information to buildings, for example.[8]

The CityGML models also introduces the Level of Detail (LoD) concept which presents the possibility to generalise CityGML features from a very detailed to a less detailed description. LoD concept of the building model of CityGML is depicted in Figure 2.

Many efforts have been made to achieve a higher-level of geometric and semantic granularity in the modules, especially in the building and core module. The LoD differentiates a Geometrical and a Semantical Level of Detail, which are separately defined for the interior and exterior features of a building. Therefore, the features of the real world are modelled, and two different types of representation are extracted. The combination of these types of representation provides even more detailed information about the Level of Details, a better description of the interior Level of Detail, a broadening of the opportunities for indoor modelling, and a better assignability to all other modules represented in CityGML [17]. CityGML can be used in COGITO to model the structures and components of the exterior and interior of the areas of interest for a construction project.

[7] https://vc.systems/en/explore/technical-articles/citygml/

[8] https://www.hel.fi/static/liitteet-2019/Kaupunginkanslia/Helsinki3D_Kalasatama_Digital_Twins.pdf

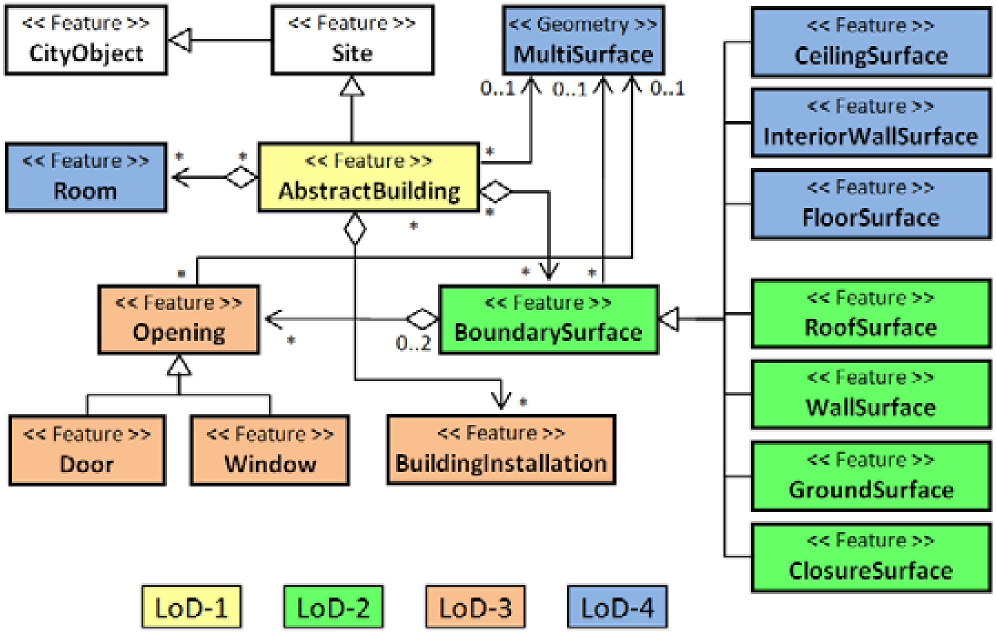

<< Feature >> CityObject
<< Feature >> Site
<< Geometry >> MultiSurface
<< Feature >> CeilingSurface
<< Feature >> InteriorWallSurface
<< Feature >> FloorSurface
<< Feature >> Room
<< Feature >> AbstractBuilding
<< Feature >> Opening
<< Feature >> BoundarySurface
<< Feature >> RoofSurface
<< Feature >> WallSurface
<< Feature >> GroundSurface
<< Feature >> ClosureSurface
<< Feature >> Door
<< Feature >> Window
<< Feature >> BuildingInstallation
0..1
0..1
0..1
0..2
*
LoD-1
LoD-2
LoD-3
LoD-4

Figure 2 – cityGML Building model [18]

3.1.2 Relevant Ontologies

3.1.2.1 DICO Ontologies

Digital construction (DICO)[9] ontologies are a suite of ontologies (see Table 1) that aim to capture the relevant objects and properties (relationships and attributes) that can be referred by people or systems during the management and execution of construction or renovation projects. They include physical and spatial entities, temporal regions, information contents, agents, activities, and groupings of objects. The DICO ontologies have been completed in the Linked Data and Ontologies work package of BIM4EEB project[10].

Table 1 – Ontologies in the DICO suite

Ontology	Namespace	Description
Entities	https://w3id.org/digitalconstruction/Entities	Identifiable entities with classifications, breakdowns and groupings
Processes	https://w3id.org/digitalconstruction/Processes	Activities, capabilities, constraints and variables
Agents	https://w3id.org/digitalconstruction/Agents	Actors and stakeholders, and their relations and contracts
Information	https://w3id.org/digitalconstruction/Information	Information content entities including designs, plans, events, and issues
Contexts	https://w3id.org/digitalconstruction/Contexts	Multi-contexts data: planned/actual, as-designed/as-built, levels of detail
Variables	https://w3id.org/digitalconstruction/Variables	Variables and constraints to support the representation of incomplete plans and management of changes

[9] https://digitalconstruction.github.io/
[10] https://www.bim4eeb-project.eu/

Occupant Behavior	https://w3id.org/digitalconstruction/OccupantBehavior	Occupant behaviour and comfort
Indoor Air Quality	https://w3id.org/digitalconstruction/IndoorAirQuality	Indoor air quality
Building Acoustics	https://w3id.org/digitalconstruction/BuildingAcoustics	Building acoustics
Energy Systems	https://w3id.org/digitalconstruction/EnergySystems	Energy systems of buildings
Building Materials	https://w3id.org/digitalconstruction/BuildingMaterials	Building materials
Lifecycle	https://w3id.org/digitalconstruction/Lifecycle	Evolution of information over construction lifecycle and refinement through LOD levels

The top-level organization of the DICO ontologies are provided by the Basic Formal Ontology (BFO). It divides the entities into two classes, Occurrent (things taking place in time, such as processes) and Continuant (things taking place in space, such as physical entities or spatial regions). From the perspective of construction management, Activity - a subclass of Process - captures the intentional efforts of an Agent. An Agent can be a Person or an Organization and can have Capabilities and assume Roles. The construction process is characterized by a set of Information Content Entities, such as Designs, Plans, Contracts and Issues.

The ontologies use or are aligned with well-known existing ontologies, as shown in Figure 3 (import relations are shown as solid and reference relations as dashed lines). The BFO is imported as the top-level ontology, the BIM models can be provided in ifcOWL (see section 3.1.2.2) or BOT (see section 3.2.1.2), and concepts from other ontologies such as OWL-Time (see section 4.2.1), PROV-O, FOAF, QUDT, SSN/SOSA (see section 5.2.1.2), and SAREF (see section 5.2.1.1) are used to the extent required.

Although BIM4EEBs and COGITO's scopes are not identical, both deal with concepts that are relevant to construction projects. That is why DICO ontologies can potentially be a very useful resource for COGITO.

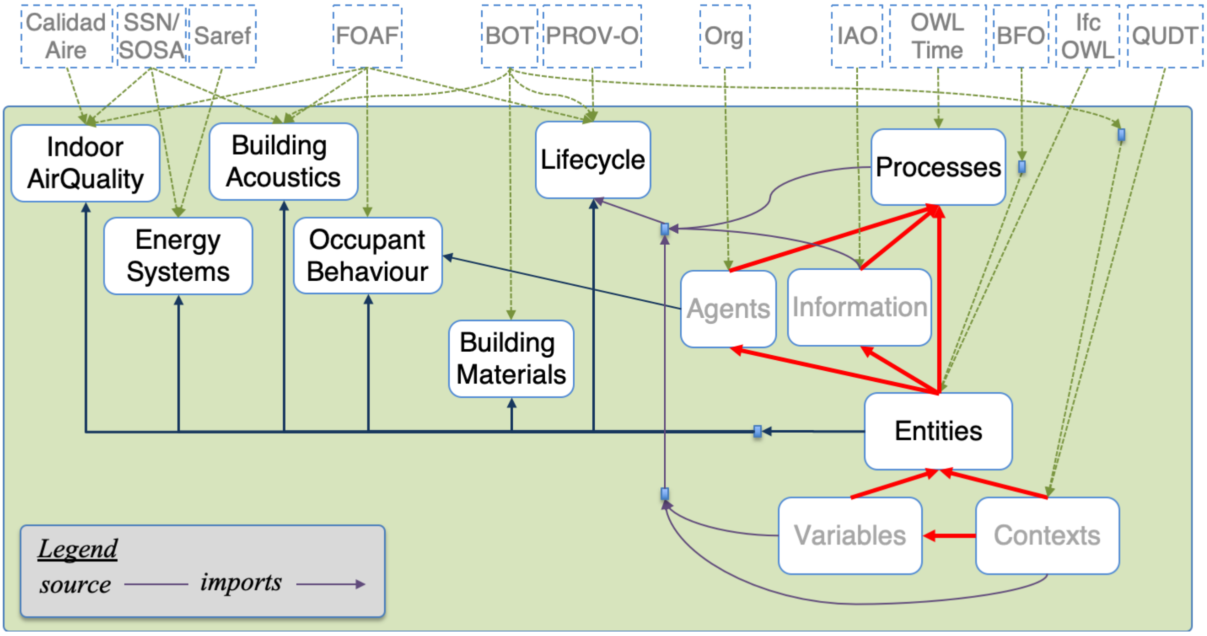


Figure 3 –External ontologies related to DICO ontologies[9]

3.1.2.2 ifcOWL

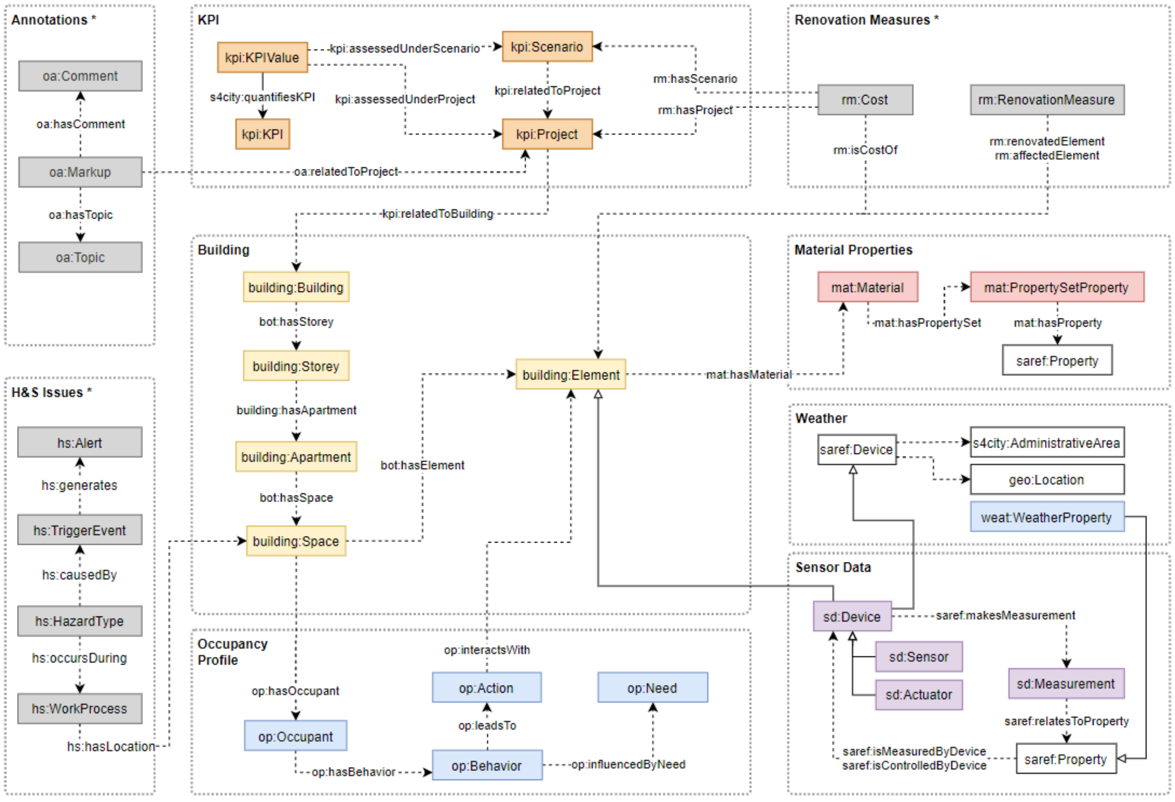
Industry Foundation Classes (IFC) are an open specification for Building Information Modelling (BIM) data[11]. ifcOWL is an OWL DL representation of the IFC schema[12]. Using ifcOWL, which represents the IFC data as directed labelled graph, allows to link building data to related data sets such as material data, GIS data, and sensor data. Data management and exchange benefit from these linked data sets.

ifcOWL is relevant to COGITO because it allows the specification of building life-cycle relevant data by open linked data standards. IFC is also oriented to the representation of construction data that may contain geometric and non-geometric data about the construction project.

The complexity of the data model allows to exchange/store rich construction-site data sets. It must be evaluated though if its high complexity benefits or hinders COGITO towards developing an efficient data exchange format.

3.1.2.3 BIMERR ontology network

The BIMERR ontology network[13] represents the semantic models that describe the different aspects of building renovation processes (e.g., energy efficiency, occupancy, building information models, etc.). Such models are defined following a modular approach, that is in the shape of a network, in which each domain (KPI, materials, building, etc.) could be reused independently (See Figure 4 and Table 2).



* **NOTE:** Model still in development

Figure 4 – The BIMERR ontology network[14]

[11] https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/

[12] https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/

[13] https://bimerr.eu/bimerr-tools/bimerr-ontology-network/

[14] https://bimerr.iot.linkeddata.es/

Table 2 – Ontologies in the BIMERR network

Ontology	Namespace	Description
Occupancy Profile	http://bimerr.iot.linkeddata.es/def/occupancy-profile	Occupants' behaviour inside buildings
Sensor Data	http://bimerr.iot.linkeddata.es/def/sensor-data	Data from sensors located inside buildings
Key Performance Indicator	http://bimerr.iot.linkeddata.es/def/key-performance-indicator	Key Performance Indicator information related to building renovation works
Weather	http://bimerr.iot.linkeddata.es/def/weather	weather data
Building	http://bimerr.iot.linkeddata.es/def/building	building data
Material Properties	http://bimerr.iot.linkeddata.es/def/material-properties	properties needed to describe building elements
Annotation Objects	http://bimerr.iot.linkeddata.es/def/annotation-objects	annotations and extra information attached to building elements
Information Objects	http://bimerr.iot.linkeddata.es/def/information-objects	files and documents attached to building elements
Renovation Process	http://bimerr.iot.linkeddata.es/def/renovation-process	construction processes in a building renovation project
Metadata	http://bimerr.iot.linkeddata.es/def/metadata	annotation properties to support the ontology to data model transformation

Since BIMERR[15] is an ongoing project, we will have to pay attention to additions and updates before deciding on the suitability of its ontologies to the COGITO project.

3.2 Building

3.2.1 Relevant Ontologies

3.2.1.1 SAREF4BLDG

SAREF4BLDG[16] extends the SAREF ontology (see section 5.2.1.1) by integrating devices that are defined in the IFC version 4 - Addendum 1 (see section 3.1.2.2,) thus enabling the representation of such devices and other physical objects in building spaces.

SAREF4BLDG is an OWL-DL ontology with 72 classes (67 defined in SAREF4BLDG and 5 reused from the SAREF and geo ontologies[17]), 179 object properties (177 defined in SAREF4BLDG and 2 reused from the SAREF and geo ontologies), and 83 data type properties (82 defined in SAREF4BLDG and 1 reused from the SAREF ontology).

Figure 5 presents an overview of the classes (only the top levels of the hierarchy) and the properties included in the SAREF4BLDG extension. As it can be observed, the classes s4bldg:Building, s4bldg:BuildingSpace and s4bldg:PhysicalObject have been declared as subclasses of the class geo:SpatialThing in order to reuse the conceptualisation for locations already introduced by the geo ontology. The modelling of building objects and building spaces has stemmed from the SAREF ontology; in this sense, the new classes deprecate the saref:BuildingObject and saref:BuildingSpace classes. In addition, a new class has been created, the s4bldg:Building class, to represent buildings.

The concepts s4bldg:Building and s4bldg:BuildingSpace are related to each other by means of the properties s4bldg:hasSpace and s4bldg:isSpaceOf; such properties are defined as inverse properties among them. These

[15] https://bimerr.eu/

[16] https://saref.etsi.org/saref4bldg/

[17] http://www.w3.org/2003/01/geo/wgs84_pos#

properties might also be used to declare that a s4bldg:BuildingSpace has other spaces belonging to the class s4bldg:BuildingSpace.

The relationship between building spaces, devices and building objects has also been transferred and generalised from the SAREF ontology. In this regard, a s4bldg:BuildingSpace can contain (represented by the property s4bldg:contains) individuals belonging to the class s4bldg:PhysicalObject. This generalisation has been implemented in order to support building spaces to contain both building objects and devices. Accordingly, the classes s4bldg:BuildingObject and saref:Device are declared as subclasses of the s4bldg:PhysicalObject.

Finally, the class that represents building devices, namely s4bldg:BuildingDevice, is defined as a subclass of both saref:Device and s4bldg:BuildingObject. This class is a candidate for replacing the saref:BuildingRelated class.

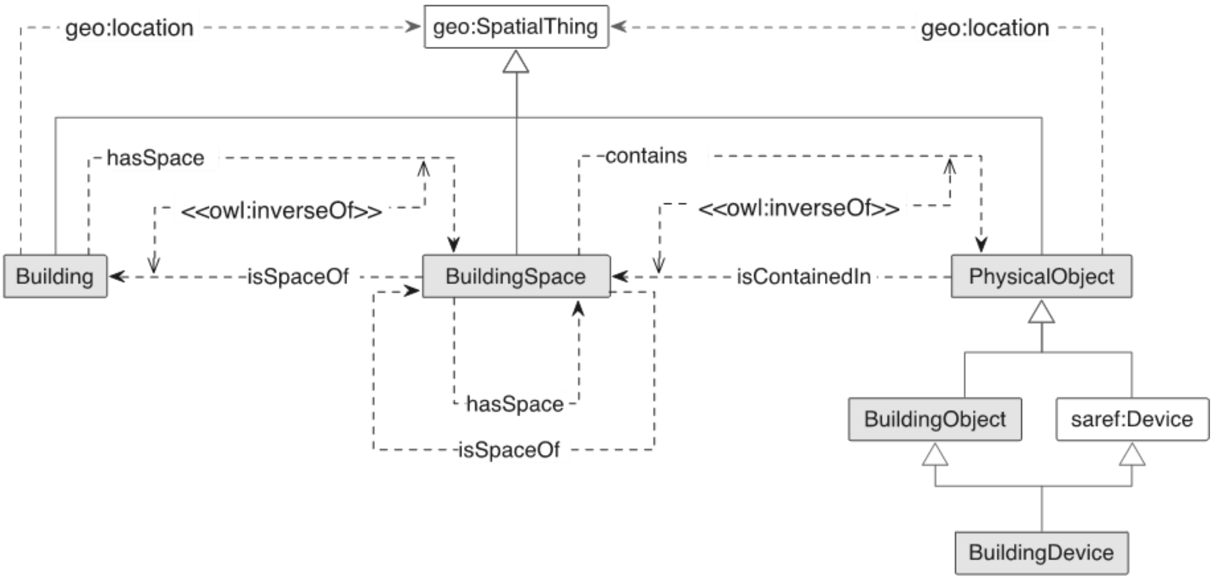


Figure 5 – General overview of the top levels of SAREF4BLDG [19]

The main contribution of this extension is the representation of the devices defined in the IFC standard and their connections to SAREF. As a result, a hierarchy consisting of 62 classes has been created, considering the subset of the IFC hierarchy related to devices and adding several classes to clarify its categorisation.

The SAREF4BLDG ontology appears to be of high relevance to COGITO, since it has been developed taking into consideration the Architecture, Engineering and Construction (AEC) and Facilities Management (FM) fields. It supports mechanisms to facilitate the exchange and interoperability of data between actors (architects, engineers, consultants, contractors, and product component manufacturers) along the different stages of the building life cycle (Planning and Design, Construction, Commissioning, Operation, Retrofitting/Refurbishment/ Reconfiguration, and Demolition/ Recycling).

3.2.1.2 W3C BOT Ontology

The Building Topology Ontology (BOT) [20] is a minimal OWL DL ontology for representing relationships between sub-elements of a building. The BOT ontology aims to combine multiple domain specific ontologies by following the common W3C principles for potential reuse. The BOT ontology is being developed by the W3C based LBDC (Linked Building Data Community) Group. This ontology presents a high-level representation of the topology of a building with spaces and stories including 3D mesh geometry of these components. Buildings, sites, stories and spaces are considered as non-physical objects that describe a spatial zone [21].

In a nutshell, the scope of BOT is to represent some specific relationships between the sub-elements of a building. A building usually consists of the building itself and several stories, rooms, and building components possibly related with each other. The object properties of the classes have specified domains and ranges so that a reasoner can infer classes automatically.

The BOT ontology is designed to allow for response to various competent questions such as: (1) how to describe adjacencies between various zones (2) how to express a building site, (3) how to express interfaces between element/element, zone/zone, or zone/element. Some specific ontologies for products, geometry, and properties are designed as domain working groups, all being aligned with BOT.

Primary concepts described in BOT are the bot:Element bot:Zone, and the bot:Interface, as shown in Figure 6. BOT Zones could be further categorised as bot:Site, bot:Space, bot:Storey and bot:Building. Zones could have adjacent zones, related by the object property bot:adjacentZone. Zones kept in a building are described by the property bot:containsZone, which could be presented by the properties bot:hasBuilding, bot:hasStorey, and bot:hasSpace.

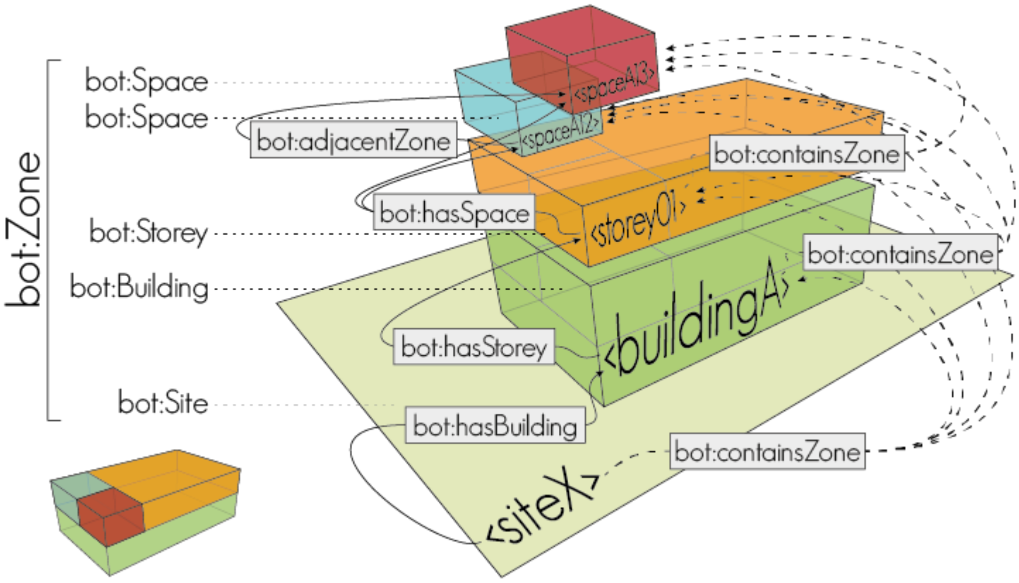


Figure 6 – Classes and relationships involved in Zones[18]

The BOT ontology is pertinent for COGITO not only for representing buildings, but also the relationships between their sub-elements, such as the zones and spaces included in them.

3.2.1.3 Brick Schema

Brick Schema[19] is an open-source ontology in RDF for describing the nature and context of different data sources in buildings. It incorporates a comprehensive class hierarchy describing the families of sensors, actuators, equipment and other building subsystems and a minimal set of relationships for describing the associations and connections between those entities. An example of building subsystem representation is illustrated in Figure 7.

Some of the notable characteristics of the Brick schema are the following:

- Extensibility: Brick classes are composed of tags, which enables the semantic reasoning and the creation of new classes in a structured manner;
- Flexibility: Brick classes are hierarchical defined so developers can express their data requirements at different levels of abstraction, ensuring proper functionality;
- Consistency: Brick classes guarantee maximum interoperability by preventing inconsistent usage of the metadata;
- Usability: The semantic-web technologies (RDF) enable better interoperability as both the schema and the data are accessible from the standard RDF/OWL tools and technologies, e.g. SPARQL, Apache Jena. They can used to support storage, querying, composition and visualization of the Brick instances.

[18] https://w3c-lbd-cg.github.io/lbd/bot/
[19] https://brickschema.org/

The Brick schema is relevant to COGITO because it is centred on the concepts of (a) Building Topology; (b) Things (HVAC, building equipment) and their relations and (c) Data Sources (gateways, controllers, databases).

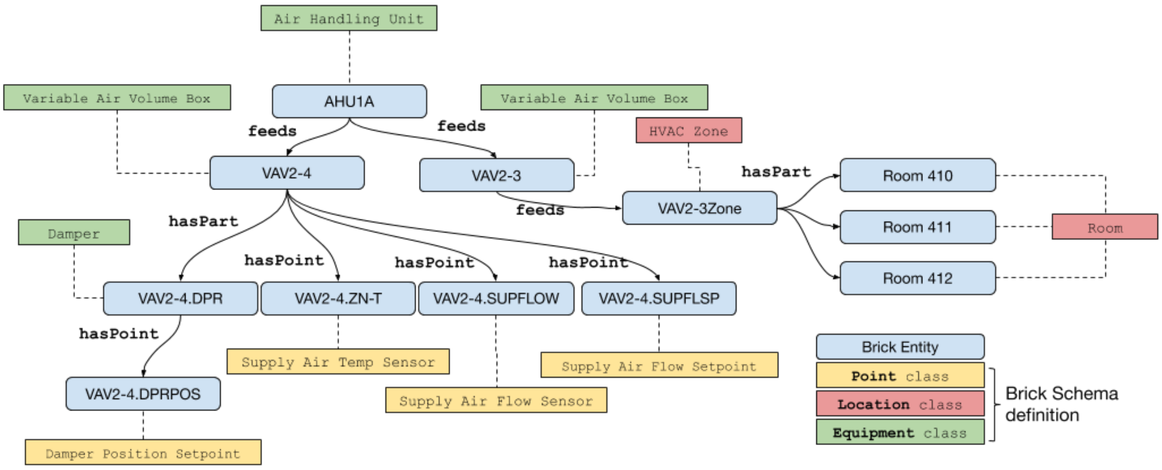


Figure 7 – Semantic representation of an AHU using the Brick Schema

3.2.1.4 Building Product Ontology

The Building Product Ontology (BPO)[20] defines concepts to describe (building) products in a schematic way. It provides methods to describe assembly structures and component interconnections and attach properties to any component without restricting their types. To allow the description of complex properties, it also contains terms for unordered, two-dimensional lists.

The BPO's scope is on the schematic description of products only, without including geometry or material compositions.

The description of assebly structures within BPO relies on three classes and two properties:

- bpo:Components, which serves as an abstract superclass of the following two classes, for ease of querying and enhanced reasoning;
- bpo:Elements, that are components that cannot – or will not – be decomposed into further subcomponents and thereby pose as the smallest and most elementar components in this structure;
- bpo:Assemblies, for components that consist of at least two sub-components (elements or other assemblies);
- bpo:consitstOf, a property that is used to indicate that an bpo:Assembly consists of the related bpo:Component(s), for enhanced reasoning – this property is defined to be transitive; and
- bpo:isPartOf, the inverse property of bpo:consistsOf that defines of which bpo:Assembly a certain bpo:Component is a part of.

Furthermore, a class for products (bpo:Product) that is also inheriting from the bpo:Component class is introduced to indicate which components can be offered as product of a vendor or manufacturer.

As shown in Figure 8, the BPO defines disjointness (red boxes) between all sibling classes with the exception of the bpo:Product class. Thus, users can define any bpo:Assembly or bpo:Element to be also a bpo:Product.

The BPO additionally provides two new relations to define that a bpo:Assembly is composed of a bpo:Entity (bpo:isComposedOfEntity) and that an bpo:Entity realises a component (bpo:realisesObject).

To describe the interconnections between product's components, the bpo:ComponentConnection class is introduced. This class is complemented by five relations:

[20] https://www.projekt-scope.de/ontologies/bpo/

- bpo:hasOutgoingConnection to define the relation between the output of a bpo:Entity and its bpo:ComponentConnection;
- bpo:ConnectsInputOIf to define the relation between a bpo:ComponentConnection and an input of a bpo:Entity it connects to;
- bpo:isConnectedTo, a chain property to link two bpo:Entity directed from the output of the domain entity to the input of the range entity;
- bpo:isConnectedFrom, the inverse property of bpo:isConnectedTo; and
- bpo:isConnectedWith, a symmetric property, to define an undirected interconnection between two components.

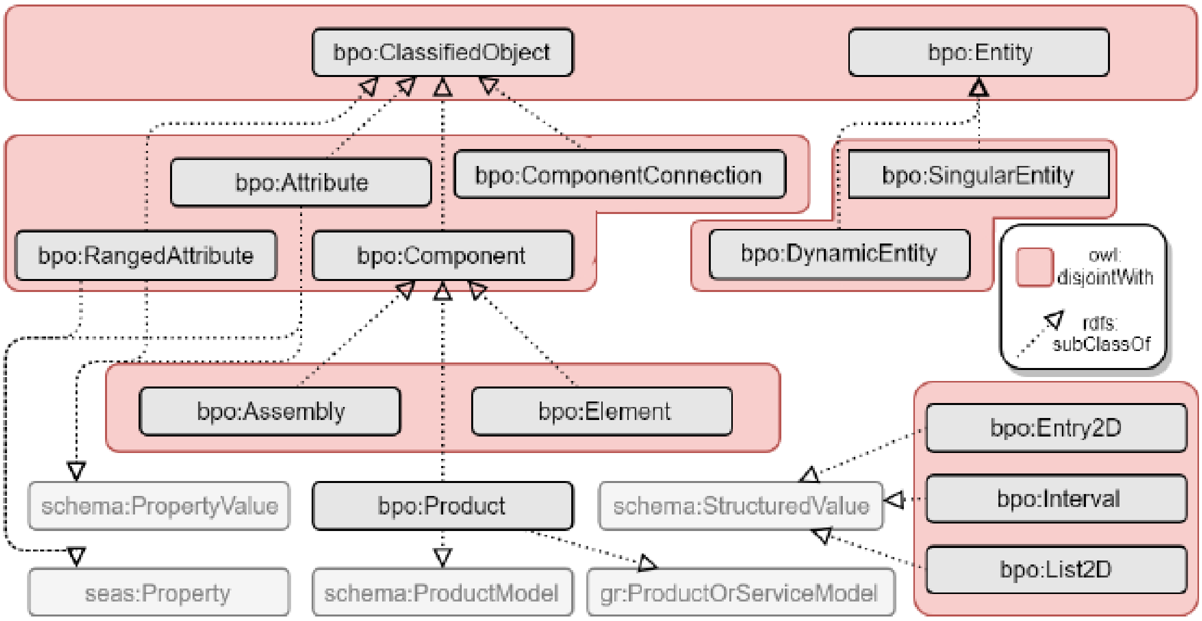


Figure 8 – Overview of the BPO Class Hierarchy [22]

The BPO ontology is relevant to COGITO since it provides means to describe building products in a schematic way, including the product structure, its properties, and its interconnections.

4 Construction Process

4.1 Relevant Data Models

4.1.1 IFC Construction Management

The IFC4 schema, introduced in Section 3.1.1.1, also contains necessary structures to describe construction management activities such as scheduling, work execution processes, task sequencing and time allocation. As Figure 9 illustrates, under the main IfcProject class, multiple IfcProduct related classes are linked to IfcProcess related classes and the construction management concepts: an instance of the abstract IfcProduct class, e.g. an IfcCurtainWall, is associated to an instance of the abstract IfcProcess class, the IfcTask class, via the IfcRelAssignsToProduct relationship.

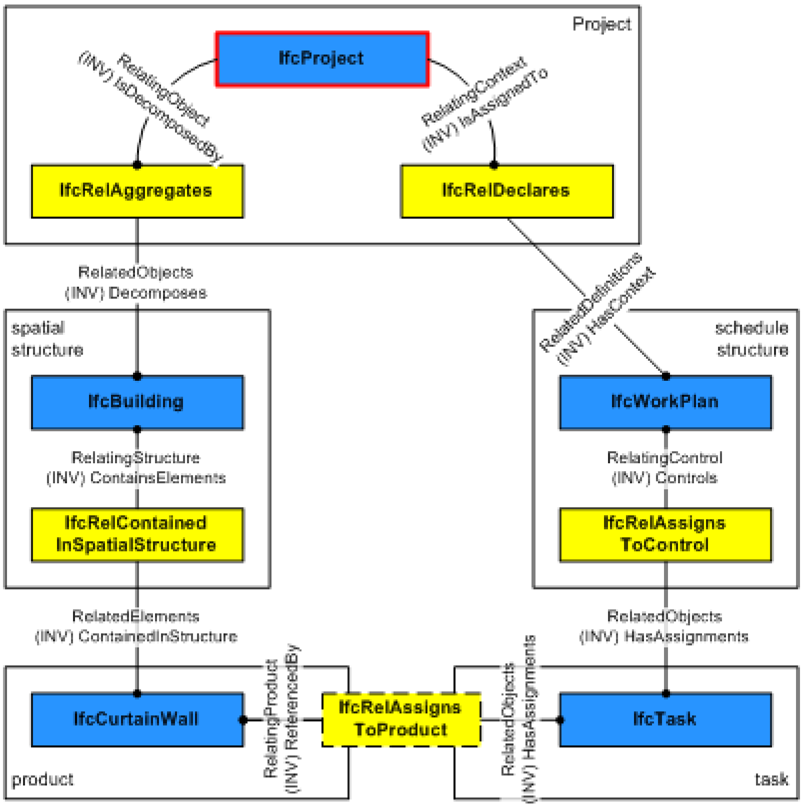


Figure 9 – Work management classes linked to product related classes

The IfcTasks are parts of IfcWorkPlans, which in turn are parts of an IfcWorkSchedule. As Figure 10 demonstrates, the IfcTasks can be stacked horizontally in the same level and split vertically into other IfcTasks (sub-tasks) at a lower level, via the IfcRelSequence and IfcRelNests relationships, respectively. The IfcTasks are also linked to timing data via the IfcTaskTime, which refers to the starting time, the ending time and the time duration of each task.

Within COGITO, the IFC Construction Process management classes can be used to support construction management operations such as process workflow management and simulation, work order management and blockchain executions and monitoring operations performed by respective COGITO tools.

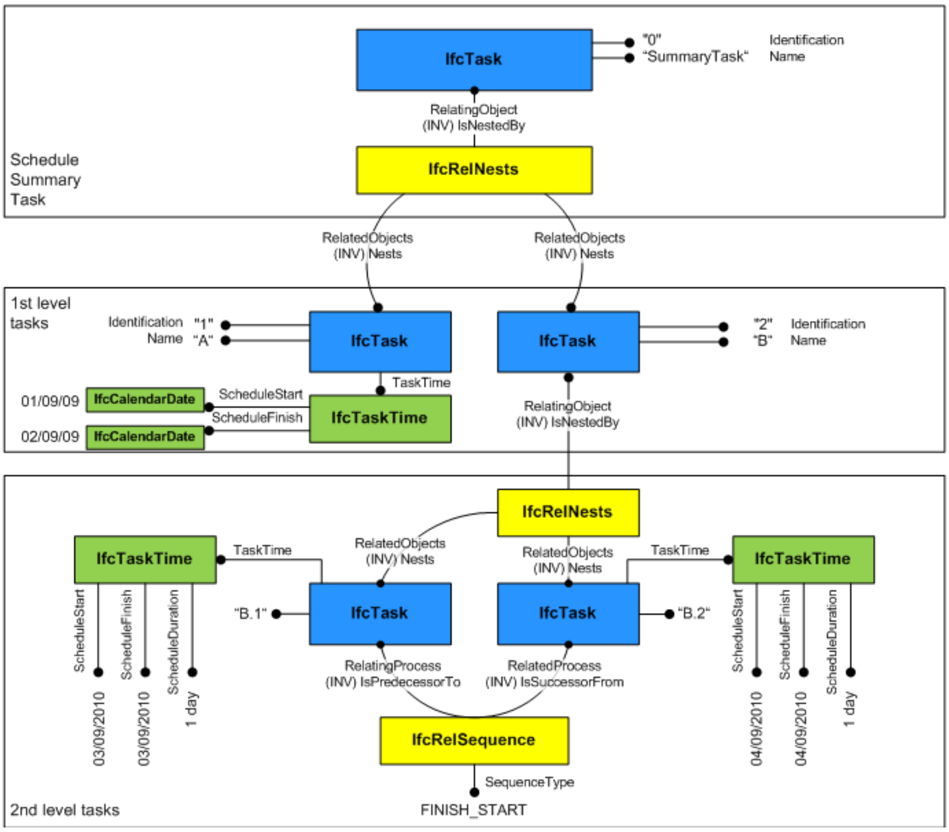


Figure 10 – Work management classes linked to task related classes

4.1.2 Business Process Model and Notation

Modern business process models are commonly represented in using the Business Process Model and Notation (BPMN) standard from OMG (c.f.[21] [22]).

The relevant concepts of the process meta model defined by BPMN are the following[23]:

- "Task", which is an atomic activity within a process flow of different types like send, receive, manual, business rule task, script or services;
- "Sub-process", which is a composed activity expressed as reference to another process;
- "Gateway", which is used to control the process flow through an exclusive gateway (exactly one path), a parallel gateway (simultaneously running several paths in parallel), an inclusive gateway (waiting of all paths to fulfil conditions) or a special event or complex gateway;
- "Event", which is something that happens during a Process and is either triggered externally or by the process itself – there are different types of events, which can be classified in start, intermediate and end events;
- "Data", which is described as input, output or as storage – the definition of the data object is highly relevant when building the digital twin of an organisation because it allows to identify requirement and expectation of each activity that will be executed in the workflow; and
- "Pool, Lanes and relations" that describe the relations between objects and processes.

The BPMN specifies the expressiveness of the modelling language in terms of graphical elements and their meanings. The formal definition of each element is not covered in the standard so each different BPMN modelling

[21] https://www.omg.org/spec/BPMN/2.0.2/PDF
[22] https://knowledge.boc-group.com/en/webinars/more-than-bpmn-2-0-bpmn-fit-for-business-practical-simple-complete/
[23] https://uk.boc-group.com/uploads/files/BOC_Poster_BPMN_20_EN_web_03.pdf

tools on the market use its own internal data model representation, however the exchange format of BPMN diagrams is defined in the BPMN specification and is worked out in the BPMN Diagram Interchange (DI) format. The format for BPMN exchanges is therefore the BPMN-DI[24].

In the BPMN Diagram Interchange[25] format the process information is divided in two sections containing the (a) process model and (b) process diagram. The process model is concerned about the process content in term of tasks, events, gateways, and their relations, hence of utmost importance when using BPMN in context of digital twinning, such as COGITO. The process diagram is concerned about the graphical representation and visual representation of the diagram, like the position of every task in the model, their size or colour and so on. We consider this as important for avoidance of vendor lock-in but not relevant for the processing of a digital twin of the organisation.

4.2 Relevant Ontologies

4.2.1 W3C Time

OWL-Time[26] is an ontology of temporal concepts, for describing the temporal properties of resources in the world or described in web pages. 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.

Figure 11 depicts the core model of the ontology, whose starting class is time:TemporalEntity with properties time:hasBeginning and time:hasEnd that link to the temporal instants that define its limits, and time:hasTemporalDuration to describe its extent. There are two subclasses: time:Interval and time:Instant, and they are the only two subclasses of time:TemporalEntity. Intervals are, intuitively, things with extent. Instants are, intuitively, point-like in that they have no interior points, but it is generally safe to think of an instant as an interval with zero length, where the beginning and end are the same.

The class time:Interval has one subclass time:ProperInterval, which corresponds with the common understanding of intervals, in that the beginning and end are distinct, and whose membership is therefore disjoint from time:Instant.

The class time:ProperInterval also has one subclass, time:DateTimeInterval. The position and extent of a time:DateTimeInterval is an element in a time:GeneralDateTimeDescription.

Four classes in the ontology support an explicit description of temporal position. time:TemporalPosition is the common super-class, with a property time:hasTRS to indicate the temporal reference system in use. time:TimePosition has properties to alternatively describe the position using a number (i.e. a temporal coordinate), or a nominal value (e.g. geologic time period, dynastic name, archeological era). time:GeneralDateTimeDescription has a set of properties to specify a date-time using calendar and clock elements. Its subclass time:DateTimeDescription fixes the temporal reference system to the Gregorian calendar.

Four additional classes support the description of the duration of an entity. time:TemporalDuration is the common super-class. time:Duration has properties to describe the duration using a scaled number (i.e. a temporal quantity). time:GeneralDurationDescription has a set of properties to specify a duration using calendar and clock elements, the definitions of which are given in the associated TRS description. Its subclass time:DurationDescription fixes the temporal reference system to the Gregorian calendar, so the time:hasTRS property may be omitted on individuals from this class. time:TemporalUnit is a standard duration which is used to scale a length of time, and to capture its granularity or precision.

Two different sets of properties are used for time:GeneralDateTimeDescription or time:DateTimeDescription, and time:GeneralDurationDescription or time:DurationDescription, because their ranges are different. For

[24] https://www.omg.org/spec/BPMN/20100501/BPMNDI.xsd
[25] https://www.omg.org/oceb-2/documents/BPMN_Interchange.pdf
[26] https://www.w3.org/TR/owl-time/

example, time:year (in time:DateTimeDescription) has a range of xsd:gYear which is a position in the Gregorian calendar, while time:years (in time:DurationDescription) has a range of xsd:decimal so that you can say "duration of 2.5 years".

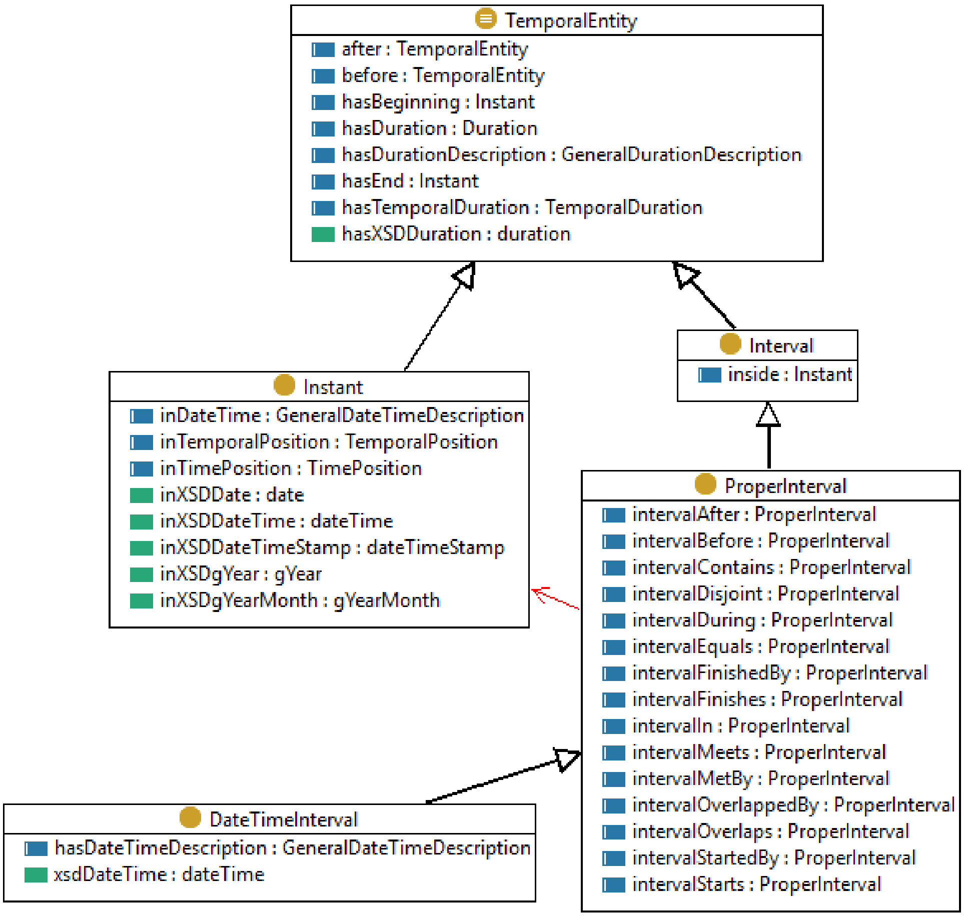

TemporalEntity
after : TemporalEntity
before : TemporalEntity
hasBeginning : Instant
hasDuration : Duration
hasDurationDescription : GeneralDurationDescription
hasEnd : Instant
hasTemporalDuration : TemporalDuration
hasXSDDuration : duration
Interval
inside : Instant
Instant
inDateTime : GeneralDateTimeDescription
inTemporalPosition : TemporalPosition
inTimePosition : TimePosition
inXSDDate : date
inXSDDateTime : dateTime
inXSDDateTimeStamp : dateTimeStamp
inXSDgYear : gYear
inXSDgYearMonth : gYearMonth
ProperInterval
intervalAfter : ProperInterval
intervalBefore : ProperInterval
intervalContains : ProperInterval
intervalDisjoint : ProperInterval
intervalDuring : ProperInterval
intervalEquals : ProperInterval
intervalFinishedBy : ProperInterval
intervalFinishes : ProperInterval
intervalIn : ProperInterval
intervalMeets : ProperInterval
intervalMetBy : ProperInterval
intervalOverlappedBy : ProperInterval
intervalOverlaps : ProperInterval
intervalStartedBy : ProperInterval
intervalStarts : ProperInterval
DateTimeInterval
hasDateTimeDescription : GeneralDateTimeDescription
xsdDateTime : dateTime

Figure 11 – Core model of temporal entities[26]

In COGITO, time-related concepts are associated to the different activities carried out during the construction phase and the state of the construction site at a certain moment. That is why this ontology can be useful for COGITO.

4.2.2 BBO

BPMN Based Ontology (BBO) [23] is also an ontology for business process representation based on the BPMN 2.0 meta-model which allows describing processes, input/output specifications, agents that perform an activity, work process to specify the process that is required to produce it, and manufacturing facilities where the process activities should be performed. The code and documentation of the ontology are also available online[27].

Figure 12 shows the main BPMN concepts reused in the BBO ontology. The starting class is bbo:Process which can be described by indicating the bbo:FlowElements that compose it. The bbo:FlowElements class has two subclasses: bbo:SequenceFlow and bbo:FlowNode. The bbo:SequenceFlow represents transitions that ensure the move from the source bbo:FlowNode to the target one. A bbo:SequenceFlow may depend on a given condition, which is represented as an instance of bbo:Expression class. The bbo:FlowNode class groups the activities that compose a process into a bbo:Activity, which refers to the work to be performed.

[27] https://www.irit.fr/recherches/MELODI/ontologies/BBO#

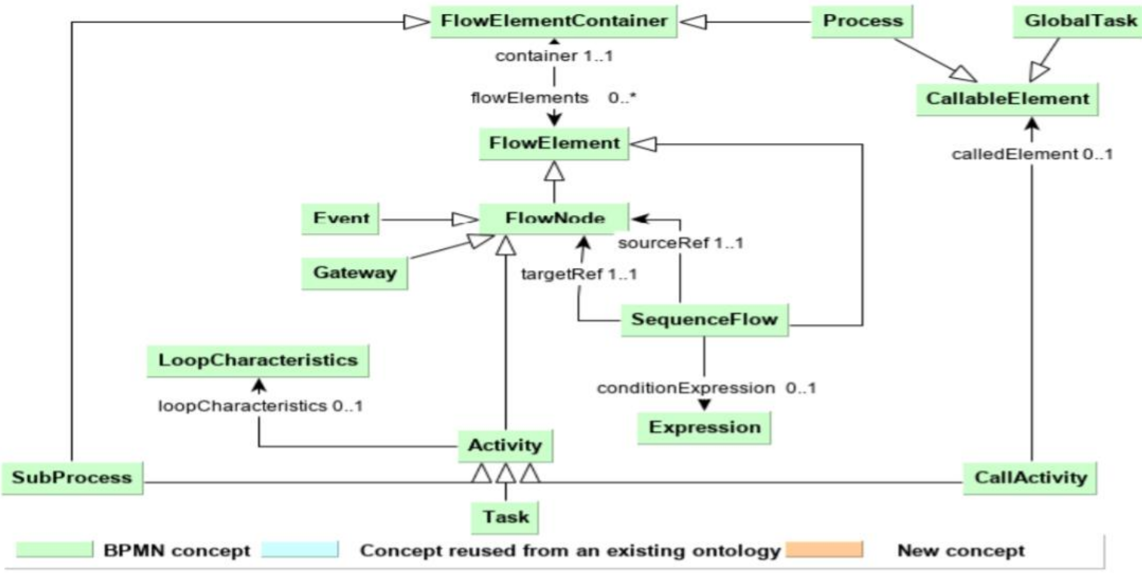


Figure 12 – Process class properties and related concepts from BPMN reused in BBO [23]

In order to represent the input and output of each activity, as shown in Figure 13, the ontology defines the bbo:InputOutputSpecification which is related to a bbo:Resource. Moreover, to represent the parameter values specifications, the BBO ontology defines the bbo:ParameterValueBinding, the bbo:Parameter and its subclasses bbo:QualitativeParameter and bbo:QuantitativeParameter, the bbo:ParameterValue, and the bbo:ParameterExpectedValue.

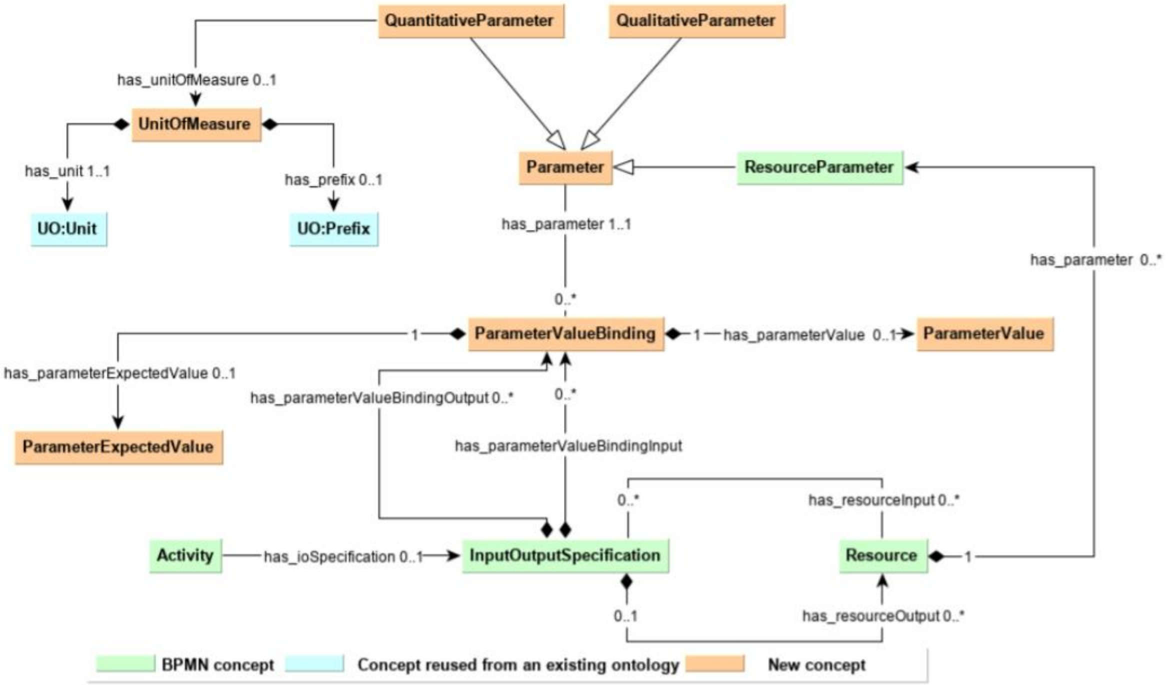


Figure 13 – InputOutputSpecification class, linked properties and classes in BBO [23]

Regarding the manufacturing facilities, the ontology defines the bbo:Station, where a particular job is performed, the bbo:Cell, which groups a set of related operations in the production flow, the bbo:Shop, where production is carried out, and the bbo:Factory which is the place where those production areas are located.

Finally, the ontology also defines work products and agents, which are types of bbo:Resource. A bbo:Agent plays a particular bbo:Role and can also have a bbo:Job.

The BBO ontology is relevant for COGITO since it allows to model business processes, including the different resources and input/output requirements involved in it.

4.2.3 DICO's Process Ontology

The Process Ontology[28] is the component of DICO that provides the concepts for activities and their relationships (flows). The relationships are represented with variables and constraints (which have their own ontology[29]).

Activity classes form a hierarchy and define to several variable properties, as shown in Figure 14:

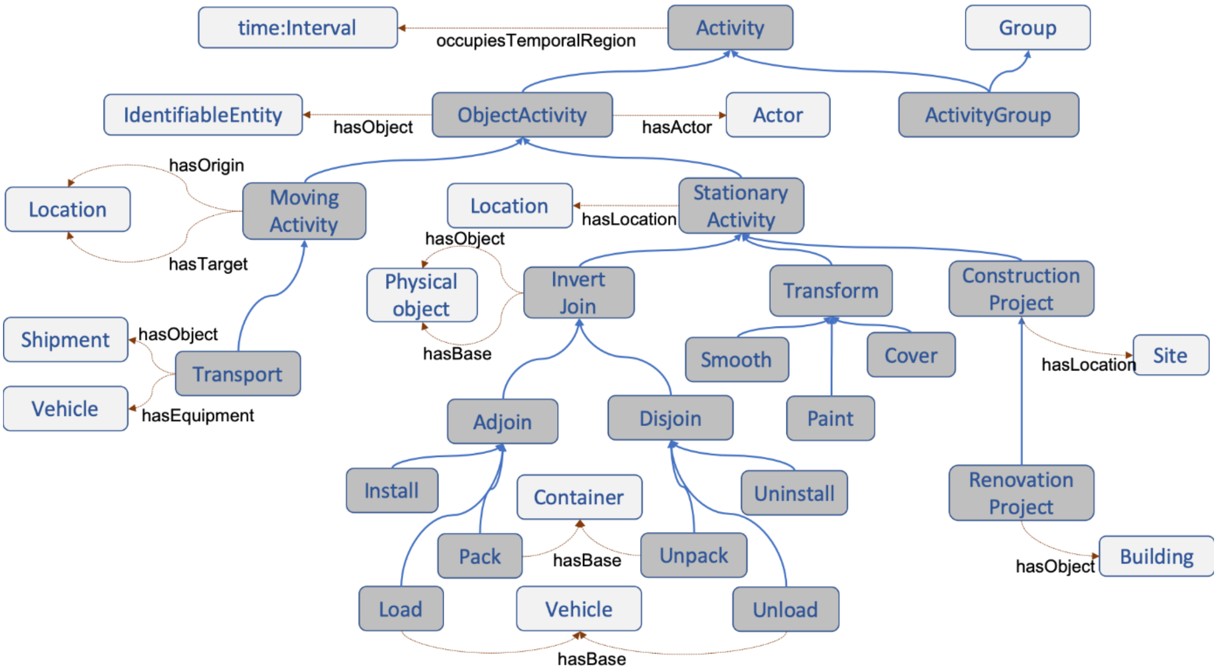


Figure 14 – Activity hierarchy in DICO's Process Ontology[28]

Each activity has flows. For instance, in the design stage, the activity-flows are mainly information entities and labour - equipment, spaces, materials, or environmental conditions do not have relevance to office activities. Moreover, there can be multiple flows of each type. An activity may require multiple different equipment, labour crews, materials or information entities for their execution.

Activities are transformations that are chained one after another based on the state changes they create. For instance, smoothing of a wall is an activity that transforms the wall surface from rough to even, and painting of the wall is an activity that requires that the surface is even. This creates a precedence constraint between the smoothing and painting activities. To represent the transformation logic of construction processes, there is a need for the conditions that an activity requires to hold for it to be executed (they can be preconditions: conditions that should hold at the start of the activity, or execution conditions: conditions that should hold all over the execution), and the effects of the execution of the activity.

Figure 15 shows examples of these types of relations. All flows have associated conditions – both preconditions, such as proper location, and execution conditions, such as reservation – and one flow has an effect that activity produces.

[28] https://digitalconstruction.github.io/Processes/
[29] https://w3id.org/digitalconstruction/Variables

From the perspective of workflow management, activities have many properties – such as different types of resources and execution times – that are gradually specified in the planning and scheduling process. The possible values are restricted by a set of constraints. The constraints can restrict the mutual values of different flows, or the conditions and effects of activities.

Taking into account that COGITO also deals with construction activities, this ontology presents a great potential for reuse in the COGITO ontology design and development.

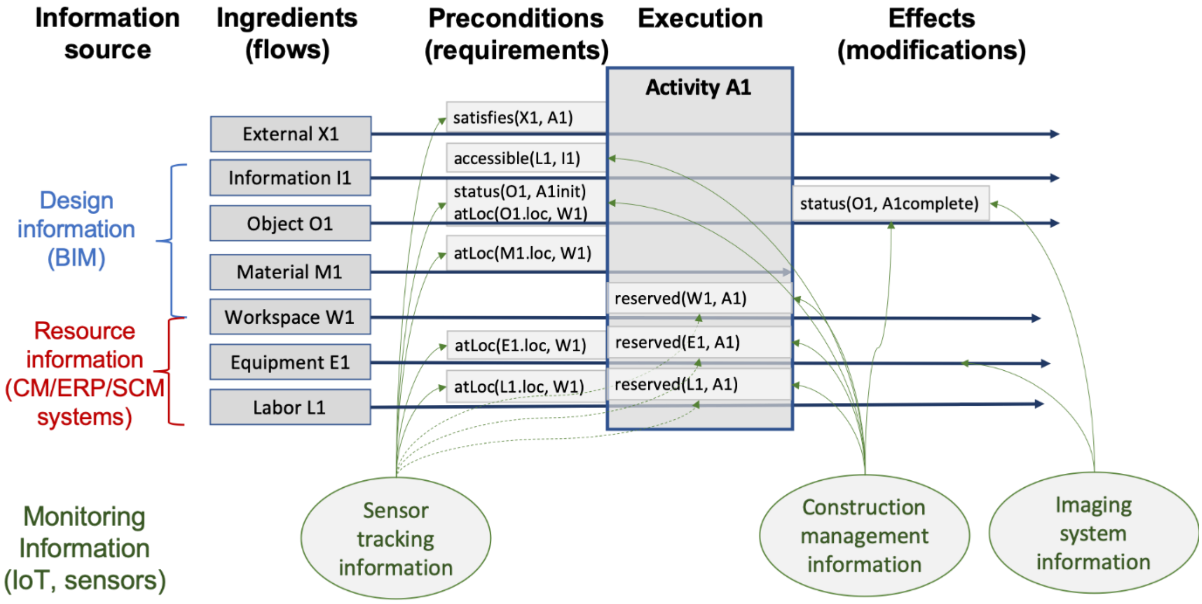


Figure 15 – Categories of flows and their relations in DICO's Process Ontology[28]

5 Reality Capture

5.1 Multi-source Visual Data

5.1.1 Relevant Data Models

5.1.1.1 E57: 3D Point Cloud Georeferenced

E57 is a data format based on the open standard for storing data produced by three-dimensional imaging systems [24]. E57 is based on the open standard ASTM E2807 [25] developed by the ASTM E57 Committee of 3D Imaging Systems. It can store point cloud data (3D coordinates and colours), as well as associated 2D imagery and core meta-data. The file structure is a hierarchical tree structure (see Figure 16), based on the XML data format. However, since it is inefficient to represent the large point clouds data in XML, at low level, the actual point data and large data block, such as images, are represented with compressed binary formats.

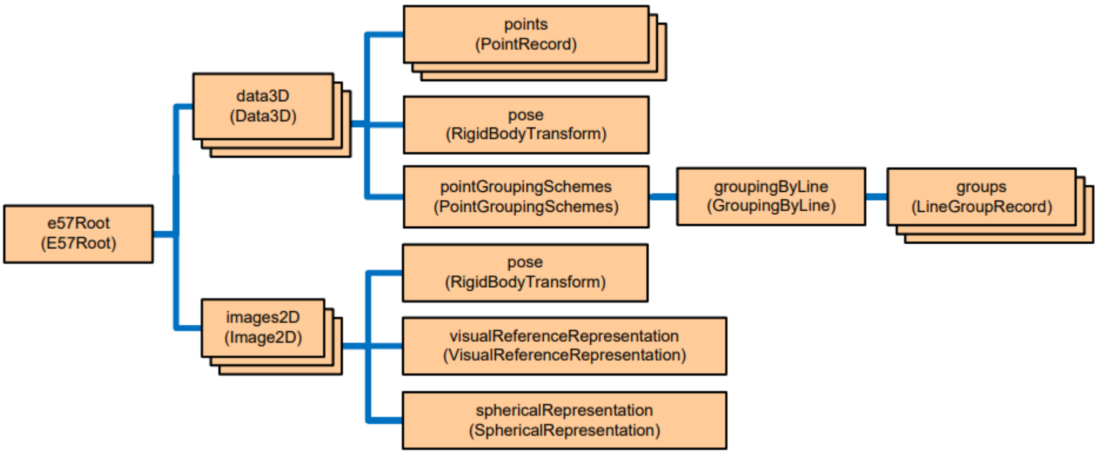


Figure 16 – E57 data structure [24]

5.1.1.2 Image Data Model by Grosky and Stanchev

Some of the earliest Image Data Models that were presented were AIR [26], VIMSYS [27] and EMIR^2 [28].

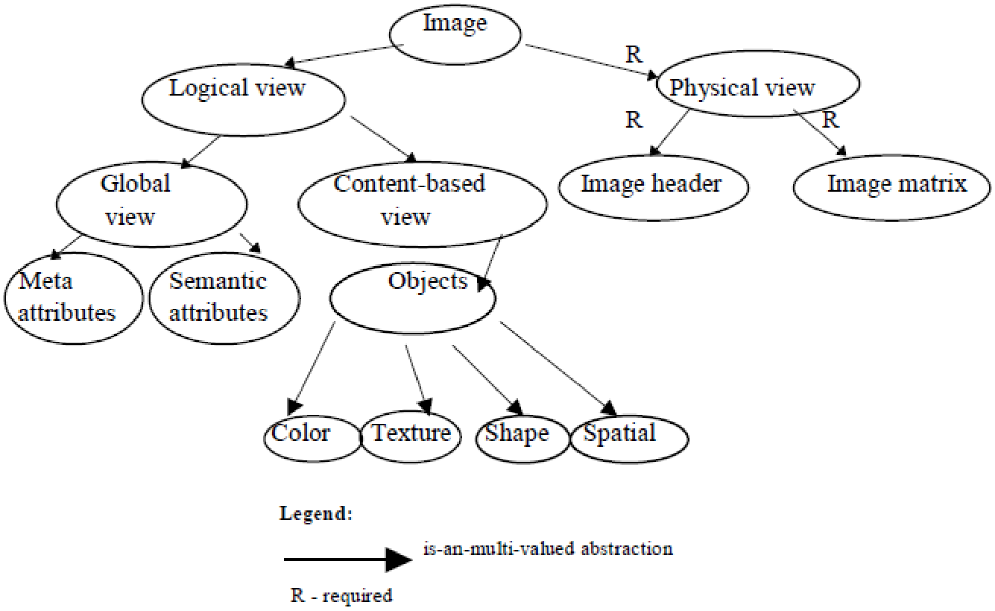


Figure 17 – Semantic Schema of the model proposed by Grosky and Stanchev [9].

Based on the systematisation of them, Grosky and Stanchev [9] proposed a new model that aims to be applicable to a wide variation of image collections and efficient because the obtained image description can be easily used for image retrieval in an image database. The data model's semantic schema is presented in Figure 17. The data model is presented at two levels, the logical and physical. The logical level contains the global description (consisted of meta and semantic attributes) and the content-based layout that contains the object features connected with colour, texture, shape and spatial characteristics. The physical level contains the image header and the image matrix.

5.1.1.3 Image Data Model by Clouard et al.

One of the Image Data Models that were presented in the later years was the one proposed by Clouard et al. [29]. According to the Data Model, for a complete definition of an image class, three different levels should be considered (see Figure 18). The first level is the physical one. It is the level of the measured signal that encodes the image data and is described by a list of effects caused by the acquisition chain components on the signal, such as noise, geometric distortion and illumination defect. The second level is the perceptive level. It is concerned with the visual rendering of the image content without any reference to objects of interest.

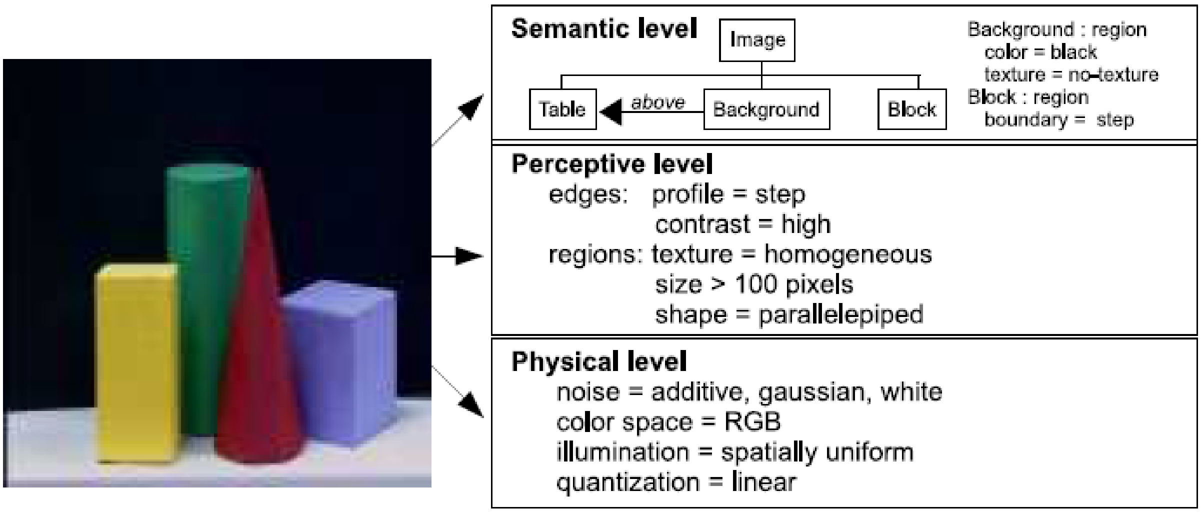


Figure 18 – An image analysed at three levels, as proposed by Clouard et al. [29].

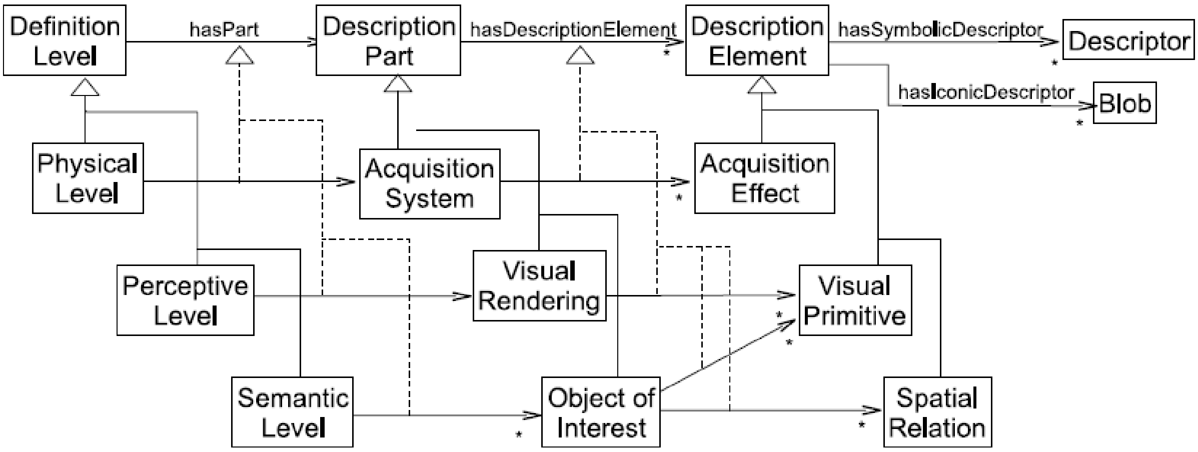


Figure 19 – UML diagram of the conceptual model for defining image classes as proposed by Clouard et al. [29].

A definition at this level is a syntactic description made from a description of visual primitives, such as region, edge, background and point of interest. The third and final level is the semantic level, which is focused on the objects of interest, understood from a phenomenological standpoint based on the visual appearance. For this reason, an object of interest corresponds to part of a scene object and in this way, several objects of interest can represent a scene object. The semantics of the scene is expressed by the discrimination between the objects of interest, by presenting information related either to the individual visual description of the objects or of their spatial relationships. Therefore, the objects are identifiable by their inner characteristics or by their spatial relations. The resulting conceptual model is presented in Figure 19, where image classes are analysed according to the three levels.

5.1.2 Relevant Ontologies

5.1.2.1 Ontology Model of the Image Object Features

In Object Based Image Analysis there are thousands of potential features that describe the objects (i.e. layer, geometry, position, texture, class-related features etc.). H. Gu et. al. [30] refer to this ontology model, developed in the eCognition software. The image object features are defined through the top–down method. There are six categories (see Figure 20) (LayerProperty, GeometryProperty, PositionProperty, TextureProperty, ClassProperty, and ThematicProperty) and each feature category can be subdivided further (i.e. the TextureProperty is divided into ToParentShapeTexture and Haralick while the Haralick subcategory is divided into GLCMHom, GLCMContrast and GLCMEntropy as illustrated in Figure below). Every subclass is shown with an “is.a” relationship.

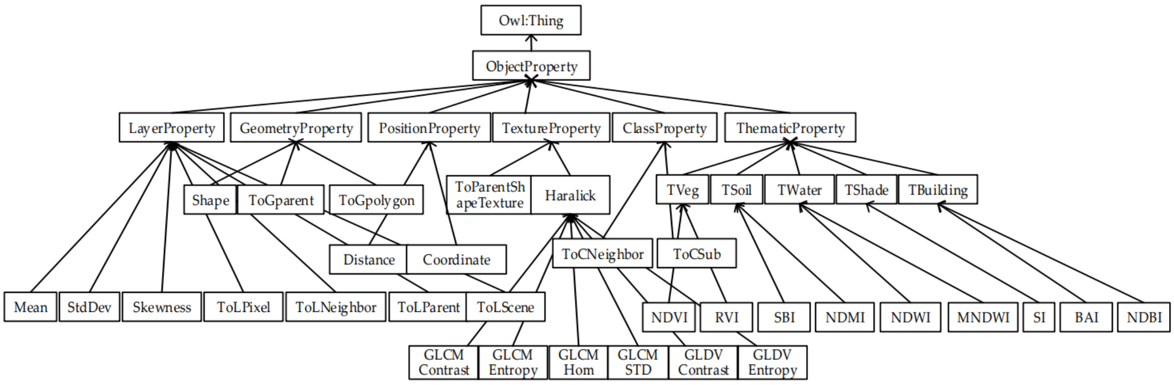


Figure 20 –Image object feature ontology [30].

5.2 Internet of Things

5.2.1 Relevant Ontologies

The Linked Open Vocabularies for Internet of Things (LOV4IoT[30]) provides a catalogue of more than 550 ontology-based projects that are addressing the IoT semantic interoperability in more than 20 domains: IoT, Wireless Sensor Networks (WSNs), Web of Things (WoT), smart home, smart energy, location, smart city, robotics, to name but a few. LOV4IoT is highly maintained with the inclusion of new references, ontology-based projects, and domains.

Despite the enormous list of existing IoT ontologies, none of them is targeting to IoT solutions for location tracking of resources in large-scale construction projects. Nevertheless, within COGITO, we aim at developing data schemas that will be easily extensible and increase interoperability with other existing solutions, so that their promotion to the appropriate standardisation bodies would be facilitated. Hence, in this section, three predominant ontologies that are supported by standardisation bodies, ETSI SAREF ontology and its extensions, W3C SSN ontology and W3C WoT Things Description ontology, and an ontology that has recently gained attention, Haystack project, have been selected to be presented.

5.2.1.1 ETSI SAREF Ontology and its Extensions

The Smart Applications REFerence ontology (SAREF), promoted by the European Commission and developed in the context of the SmartM2M Technical Committee activities at the European Telecommunications Standards Institute (ETSI), is intended to enable interoperability between solutions from different providers and among

[30] The LOV4IoT ontology catalog is available at http://lov4iot.appspot.com/

various activity sectors on the Internet of Things (IoT), thus contributing to the development of the global digital market.

SAREF explicitly specifies the recurring core concepts in the Smart Applications domain, the main relationships between these concepts, and axioms to constrain the usage of these concepts and relationships.-[31].

The SAREF ontology is intended to provide a core model for IoT that could be extended and adapted to cover specific domains. As depicted in Figure 21, the core SAREF ontology focuses on the definition of smart applications for which the main concept defined is that of device (saref:Device) that can be divided into sensors (saref:Sensor) and actuators (saref:Actuator).

SAREF allows the representation of the function(s) that a device can have by means of the concept saref:Function. A device could be designed to perform several functions. In addition, a function can have one or more commands (saref:Command) that represent the directive that a device must support to perform a certain function. Such commands might act upon a state (saref:State). SAREF also allows the representation of services (saref:Service) that make the functions discoverable, registrable and remotely controllable by others. In this ontology, the task for which a device is designed from a user perspective (for example, the task of "bake" for an oven) can also be represented by means of the concept saref:Task. The observations made by a device are represented by the concept saref:Measurement which is also related to the saref:UnitOfMeasure in which the observation is measured and the saref:Property being measured.

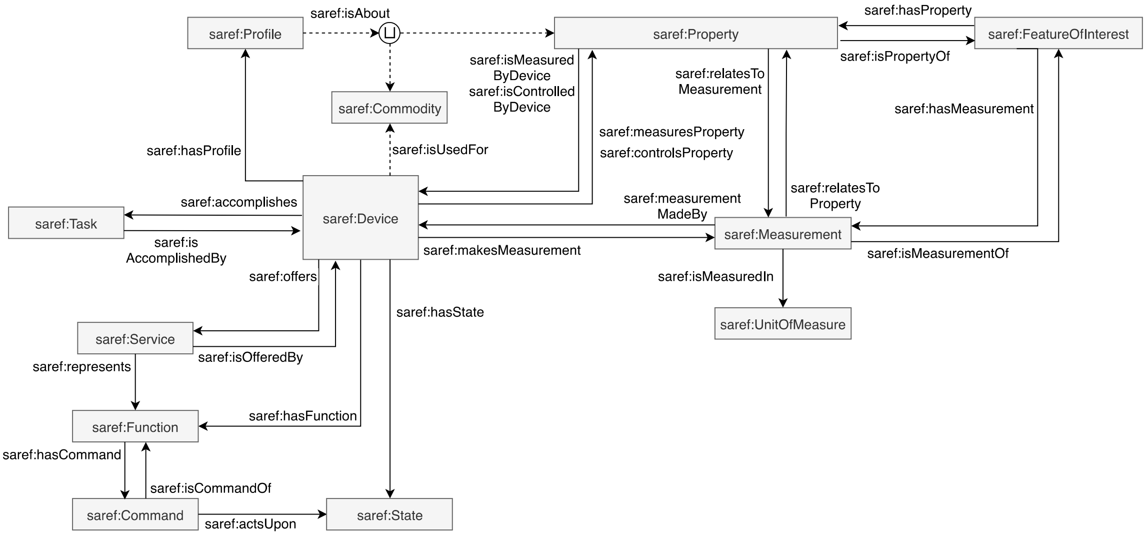


Figure 21 – General overview of the SAREF ontology [31]

Several Technical Specifications have been produced that extend the SAREF ontology to ten different domains[31], as can be seen in Figure 22, namely energy (SAREF4ENER), environment (SAREF4ENVI), building (SAREF4BLDG, see section 3.2.1.1), smart city (SAREF4CITY), industry and manufacturing (SAREF4INMA), agriculture (SAREF4AGRI), automotive (SAREF4AUTO), eHealth and ageing well (SAREF4EHAW, see section 7.1.2.1), wearables (SAREF4WEAR, see section 7.1.2.2), and water (SAREF4WATR).

[31] The SAREF ontology and its extensions are available at https://saref.etsi.org/

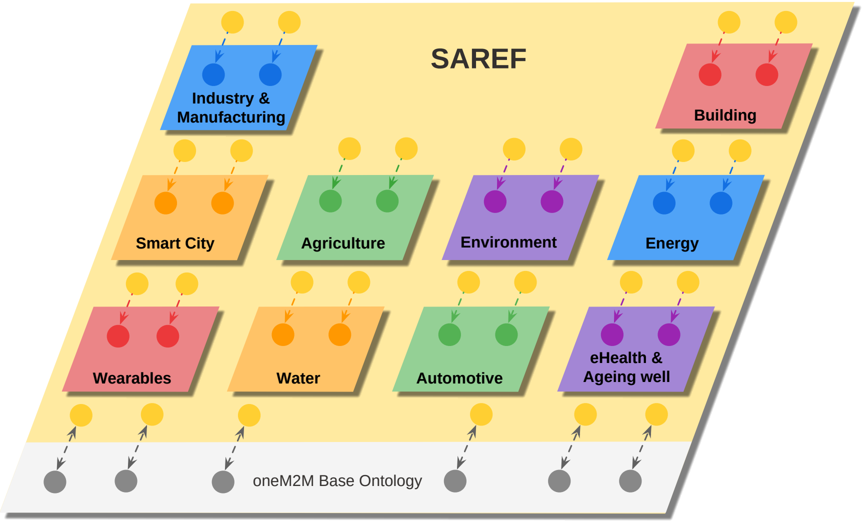


Figure 22 – The SAREF ontology and its extensions [31]

The SAREF ontology is relevant to COGITO since it is intended to provide a core model for IoT, which is key in the project.

5.2.1.2 SOSA/SSN Ontology

The SOSA/SSN ontology is a joint World Wide Web Consortium (W3C) and Open Geospatial Consortium (OGC) standard specifying the semantics of sensors, observations, actuation, and sampling.

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[32].

Figure 23 provides an overview of the classes and properties in the core of the SSN ontology, showing how the three components ("sensors and observations", "samplings and samples" and "actuators and actuations") use the same pattern.
To describe sensing acts, SOSA provides the concept sosa:Sensor that makes observations (sosa:Observation) about some observable properties (sosa:ObservableProperty) of a feature of interest (sosa:FeatureOfInterest).

To record the results of sensing acts, SOSA presents the concept sosa:Sampler that makes sampling (sosa:Sampling) of some sosa:FeatureOfInterest to produce samples (sosa:Sample) or results (sosa:Result).

To describe the ability of using actuators for some actions, SOSA includes the concept sosa:Actuator that makes some actuation (sosa:Actuation) on some actuation property (sosa:ActuationProperty) of a sosa:FeatureOfInterest.

Conceptually, an observation, sampling, or actuation can be realised as implementations (parts) of a procedure (sosa:Procedure), which is a reusable workflow, protocol, plan, algorithm, or computational method that can be used, among others, to specify how an observation activity has been made. Using SSN, a sosa:Procedure can be linked to some description of the ssn:Input and ssn:Output of these procedures.

[32] https://www.w3.org/TR/vocab-ssn/

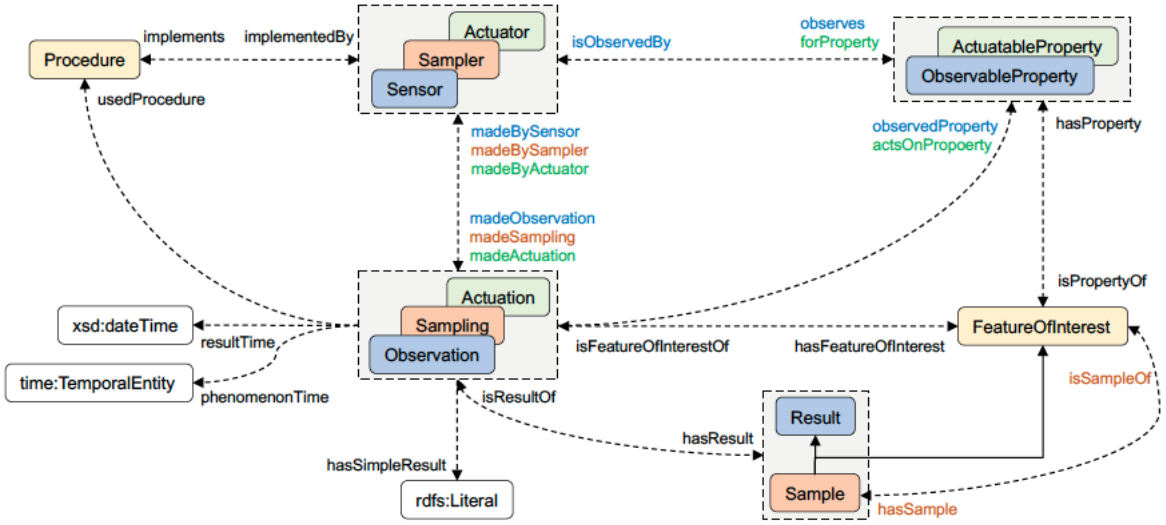


Figure 23 – Overview of the core structure of the SOSA/SSN ontology [32]

Similar to other ontologies in this survey, SSN/SOSA can provide useful concepts to represent information related to the IoT in COGITO.

5.2.1.3 Haystack Model

Haystack[33] is an open-source project that aims to provide a standard methodology to organize the building data using a tagging model and semantic web technologies. Haystack is in charge of providing annotations to static data, e.g. building systems, sensors, actuators and devices, and to dynamic data, e.g. data points and stream data. Haystack is centred on the following four basic models:

- The Core model, which includes the tagging meta-model definition, the structure, the time and the unit representation;
- The Data Structure and Text Format model, which includes the various data structures such as grids, scalar data types and external data formats;
- The REST API model, which is designed to support the exchange of haystack tags between different applications;
- The System and Equipment model, which provides the standardized representations of tags for a set of predefined MEP equipment such as Chillers, Boilers, Lighting, Pumps etc.

The fact that Haystack provides a standardised way to describe the static and the dynamic data of the installed MEP equipment, makes it relevant to COGITO. The COGITO Digital Twin platform may use the Haystack annotations to populate the relevant data models.

5.2.1.4 W3C WoT Things Description Ontology

Within W3C's WoT working group, the WoT Thing Description (TD)[34] has been emerged as a thing-centric formal model and a common representation for WoT. According to W3C's WoT working group, a Thing represents an abstraction of a physical or virtual entity which interacts and participates in the WoT, while a TD describes the metadata and interfaces of Things.

The TD Information Model is built upon the following, independent Vocabularies: (1) the core TD Vocabulary, which reflects the Interaction Model with the Properties, Actions, and Events Interaction Affordances; (2) the Data Schema Vocabulary, including (a subset of) the terms defined by JSON Schema; (3) the WoT Security Vocabulary, identifying security mechanisms and requirements for their configuration; and (4) the Hypermedia Controls Vocabulary, encoding the main principles of RESTful communication using Web links and forms.

[33] https://project-haystack.org/

[34] https://www.w3.org/TR/wot-thing-description/

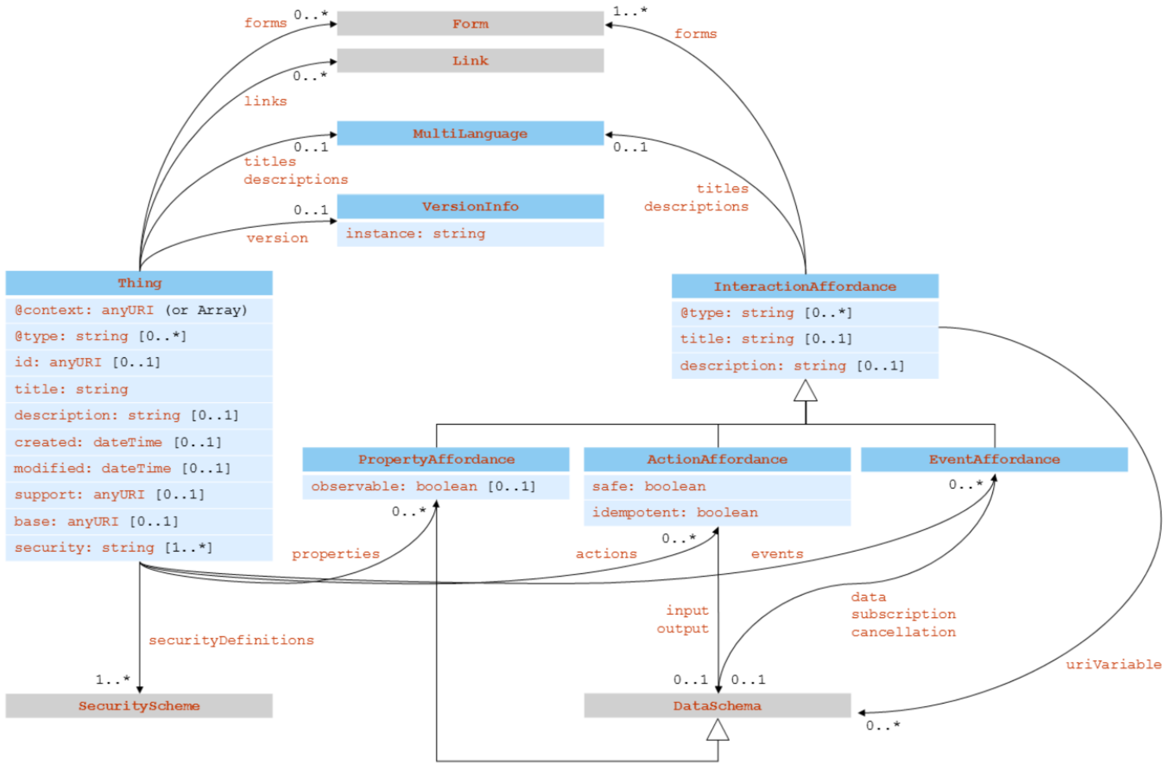


Figure 24 – WoT TD ontology: Core TD Vocabulary[34]

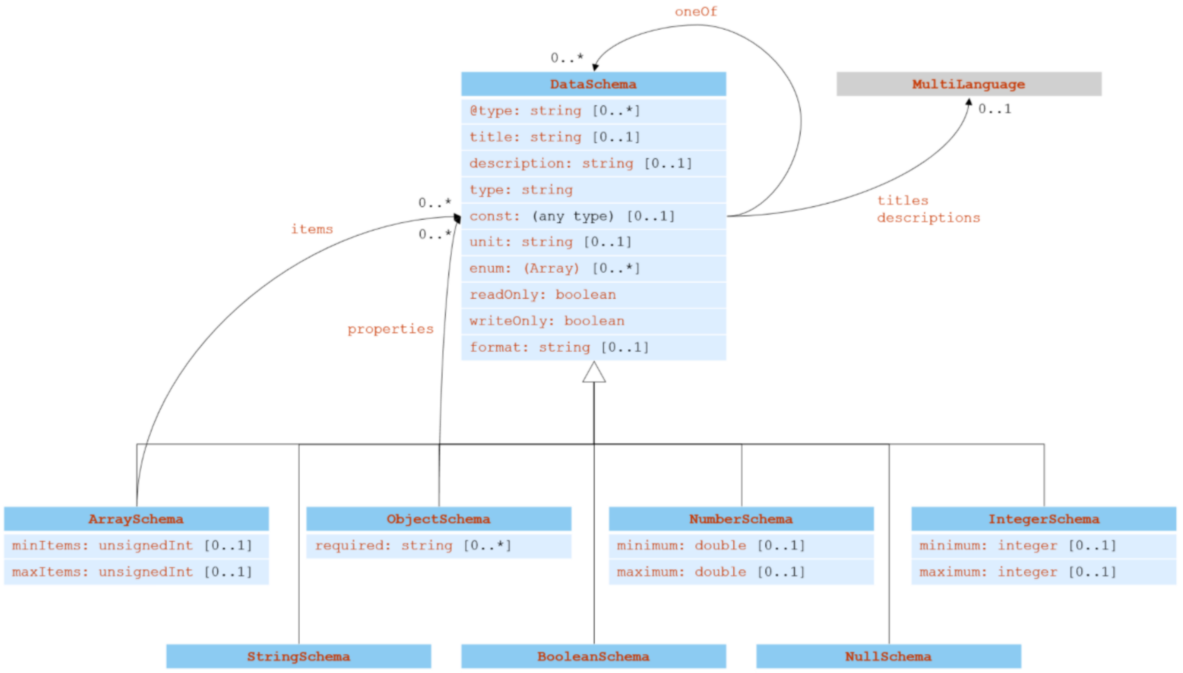


Figure 25 - WoT TD ontology: Data Schema Vocabulary[34]

The Core TD Vocabulary (see Figure 24) defines three types of Interaction Affordances (InteractionAffordance class): (1) Properties (PropertyAffordance class) is used for sensing and controlling parameters by exposing internal state of the thing, such as getting the current value or setting an operation state; (2) Actions

(ActionAffordance class) model the invocation of physical (and hence time-consuming) processes, and set the internal state of the Thing when this state is not exposed as a Property; and (3) Events (EventAffordance class) are used to describe the event sources that asynchronously push messages, thus being used for the push model of communication. Moreover, the DataSchema vocabulary, which is shown in Figure 25, presents the supported data types (e.g., ArraySchema, ObjectSchema) and enables the linked data declaration of data types. Finally, the SecuritySchema vocabulary includes well-established security mechanisms such as authentication, authorisation, and encryption with different possible levels of security configurations.

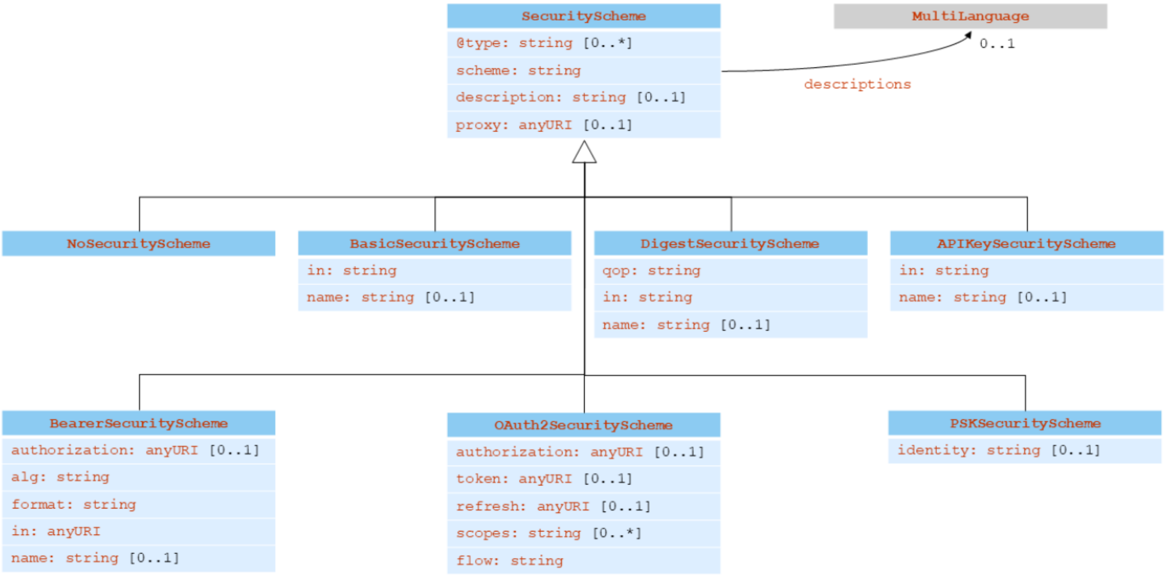


Figure 26 – WoT TD ontology: WoT Security Vocabulary[34]

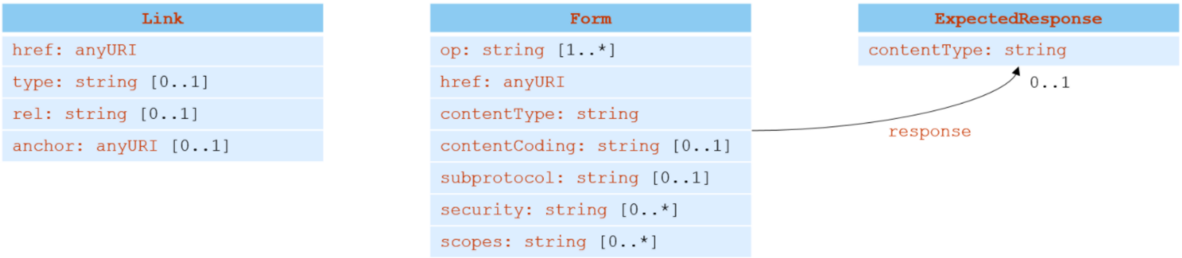


Figure 27 – WoT TD ontology: Hypermedia Controls Vocabulary[34]

WoT TD provides powerful enough metadata to capture different communication scenarios, identified by binding together URI schemes, content types and security mechanisms (authentication, authorization, confidentiality, etc.). It provides a set of interactions, based on a small vocabulary, which makes it possible to integrate diverse devices and to allow diverse applications to interoperate. Since its objective is to make the Web an interoperable platform for IoT data, the focus point is mainly on the Data Layer of IoT Oriented Criterion (IOC) with high-level schemas for describing dynamic behaviours of things; also, by modelling both the syntax and the semantics of behaviours, it supports the Findability Layer of Web Oriented Criterion (WOC), while its security schema covers the Sharing Layer.

5.3 Simulation

5.3.1 Relevant Ontologies

5.3.1.1 STATO

STATO (STATistics Ontology)[35] is a general-purpose statistical ontology which can provide coverage for various statistical process including statistical tests and analyses, information on when these tests should be applied and output from analyses including probability distributions and statistical metrics. STATO can also be used to display graphical representations of the data and describe different experimental designs used to collect data.

It is a resource that aids in the communication of scientific results created using statistical methods, allows statistical results to be linked directly to project hypotheses, and provides a framework to standardise results of statistical methods to allow groups of people with different expertise to understand analyses.

STATO was originally started in 2012 as part of the OBO Foundry ontologies[36] as an ontology to store and interrogate the results of data analysis. STATO is funded by community-driven ISA Commons-based projects and user communities and has formed various collaborations with data publication platforms (e.g. Ontobee[37]). Although STATO was originally intended to be used by researchers and practitioners in the natural and biomedical sciences, it is not limited to such fields and has since included projects from other scientific disciplines. Because of its original connection with OBO Foundry ontologies, STATO can be used to operate alongside other ontologies. This makes STATO a potentially useful resource in COGITO.

[35] http://stato-ontology.org/

[36] http://www.obofoundry.org/

[37] http://www.ontobee.org/

6 Workflow Management and Smart Contracts

6.1 Workflow Management

Our literature review did not reveal any existing data model and/or ontology relevant to the workflow management. It remains though a subject for further investigation. In this section, two data models that are supported by the tools that will be used for the Workflow Modelling and Execution services of COGITO have been selected to be presented.

6.1.1 Relevant Data Models

6.1.1.1 ADOxx-based Modelling Tools Data Models

The Workflow Modelling tool that will be used in COGITO (ADOxx-based) adheres to the BPMN standard format to model the workflows. The designed BPMNs will then be send to the Workflow execution engine (I3D System) to monitor the progress of the workflows. The BPMN standard format has been previously introduced in the description of the process format. The workflow reuses the same format because it is the result of the construction process refinement from an abstract business level to a more detailed executable level.

Additionally, a specific model type will be used to model KPIs and associate them to specific workflow activities. The KPIs meta-model will be based on concepts of the balanced scorecard[38], extended with a data model-type that allow to specify how the KPIs are retrieved and calculated.

The KPI model allows to define KPIs and Goals with their relations and group them in specific perspective:

- Perspective group similar KPIs, like grouping all “Financial” indicators or all “Time” or “Quality” dependent indicators;
- Goals and sub-goals describe the objective to be achieved;
- KPIs describe measurable data sets that assess in combination with the indicator context – plan value, real value, thresholds, type of thresholds and meta data about the indicator, if the corresponding goal can be achieved or not.

The data calculation model is instead used to describe the data that a KPI is using. It is composed of Metrics and Data access items, including dependencies between them:

- Metrics, represent a dataset in a specific format and contain information on how the value of a data is calculated, using as inputs sub-metrics and data access items;
- Data access items, on the other side describe how a dataset is retrieved from an external system that can be as example a remote service, a sensor, a database or even an Excel sheet. In this context, the data access item is strongly dependent on the ADOxx Olive Microservice Framework that is responsible to provide the features to access such external system.

Being based on BPMN standard, the workflow model use BPMN-DI as exchange format while the KPI model rely on a specific JSON and XML structure provided by the ADOxx platform.

6.1.1.2 I3D Data Model

Two of the tools available through the COGITO project implementation are Work order definition and monitoring (WODM) tool and Work order execution assistance (WOEA) tool. Both will be based on the i3D framework technology developed by NT. The framework uses a proprietary data model developed to support all i3D needs (see Figure 28). It encompasses work orders, including required activities, instructions or other additional information, actual status, and relation to the corresponding workflow, using it to guide and help workers while performing their respective tasks.

[38] Kaplan, Robert S., and David Norton (1992). "The Balanced Scorecard: Measures that Drive Performance." Harvard Business Review 70, no. 1.

Primary information input is represented by workflows. Approved workflows can be transformed into more detailed work orders. Work orders contain all necessary information needed by its providers, usually workers. Work order actions can be unitary or composed of service tasks. Each action can have multiple preconditions defined. Hotspots and info resources encompass points of interest or additional information regarding particular action. Each completed work order must have a result. Hotspots, info resources and work order results can be accompanied by additional multimedia content such as document files, pictures, voice recordings or even videos.

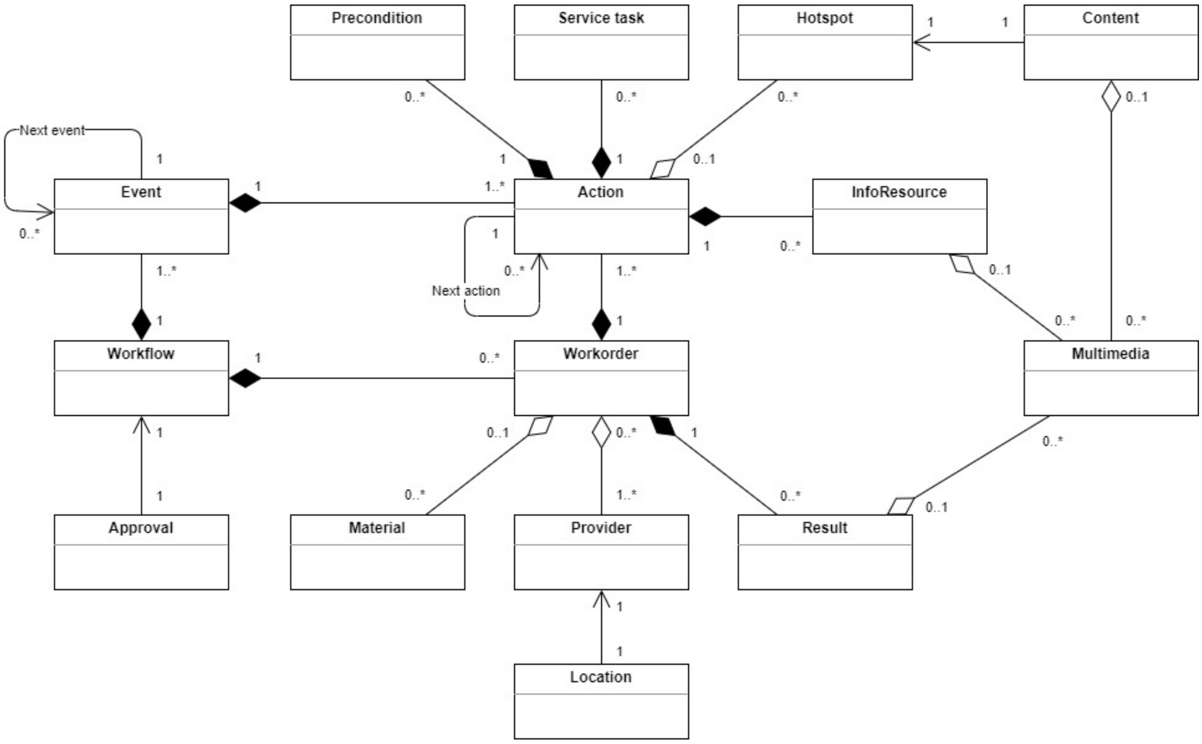


Figure 28 – Conceptual data model of i3D solution

This data model is expected to provide appropriate features to represent workflows in COGITO.

6.2 Smart Contracts

6.2.1 Relevant Data Models

6.2.1.1 bcBIM Data Model

A novel approach to tackle BIM data auditing, security and provenance is facilitated through bcBIM. More specifically, bcBIM is a model that is implemented on blockchain that can provide historical traceability of modifications through timestamps [33].

As it can be seen in Figure 29, every block consists of a Block Header that includes the previous Block Hash and the BIMHASH that ensures the integrity of data. Then follows the main body of Data that in the case of bcBIM is the LinkofData that can find the location of BIMDATA on the local storage media, Timestamp marks the timeseries of the block, BIMINDEX includes information related to the BIMDATA such as keywords and object and model numbers and finally Signature ensues the authenticity of the block.

The application of the bcBIM model in the design phase can enable collaborative modification, allowing for an easier detection of design conflicts. Additional advantages can be gained in the construction phase as it can provide reliable recording of financial activities in supply chain, credit reporting and ownership management regarding building materials, bookkeeping automation in logistics supply chain and improve resource sharing of physical assets such as heavy machinery.

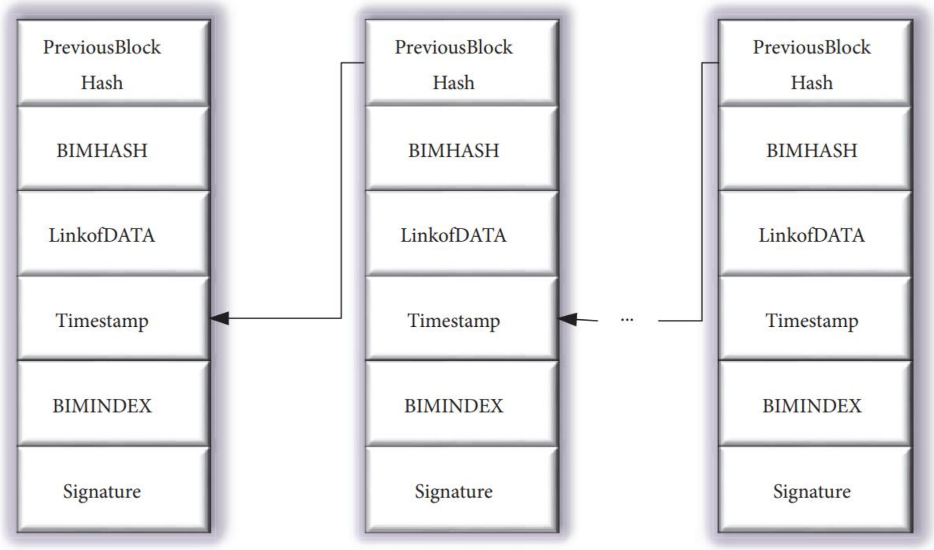


Figure 29 – bcBIM blockchain structure [33].

6.2.2 Relevant Ontologies

6.2.2.1 EthOn and its Contract Extension

The purpose of the EthOn ontology[39] is to model Ethereum[40], a particular implementation of the blockchain technology, as a State Transition System.

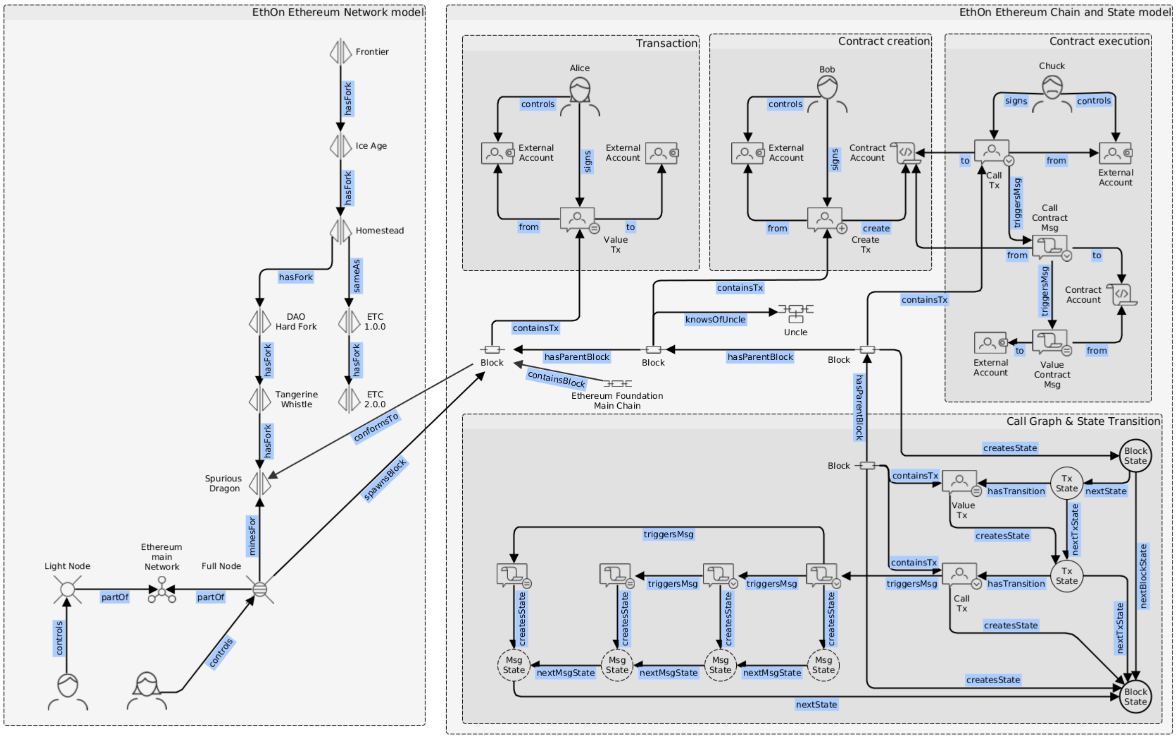


Figure 30 – How EthOn models Ethereum (simplified)[41]

[39] https://ethon.consensys.net

[40] https://ethereum.org/

[41] https://consensys.github.io/EthOn/EthOn_spec.html

Ethon supports concepts such as blocks, accounts, transactions, contract messages, nodes, etc. It is accompanied by extensions to represent the behaviour of smart contracts and Ethereum-based currencies (ERC-20 tokens).

A simplified model of the EthOn ontology is depicted in Figure 30. The model is divided into two parts: the EthOn Ethereum Network model, which represents the Ethereum blockchain in general terms, i.e. nodes, forks or versions, and the EthOn Ethereum Chain and State model, which represents the blocks and the transitions that exist within the block, i.e. transactions, smart contracts creation, smart contracts execution or the state transition of the blocks.

EthOn provides an extension[42] that models some aspects of smart contracts, including classes for functions (structures in smart contracts that implement functionality, just as functions or methods do in general programming), Events (smart contract reaction when a condition is met), Inputs (that represent a single parameter of a function or event) and Outputs (that represent a single return value of a function). Figure 31 illustrates how EthOn Contract extension models a smart contract. A throughout analysis of this extension will be carried out to decide on whether it can be used in COGITO.

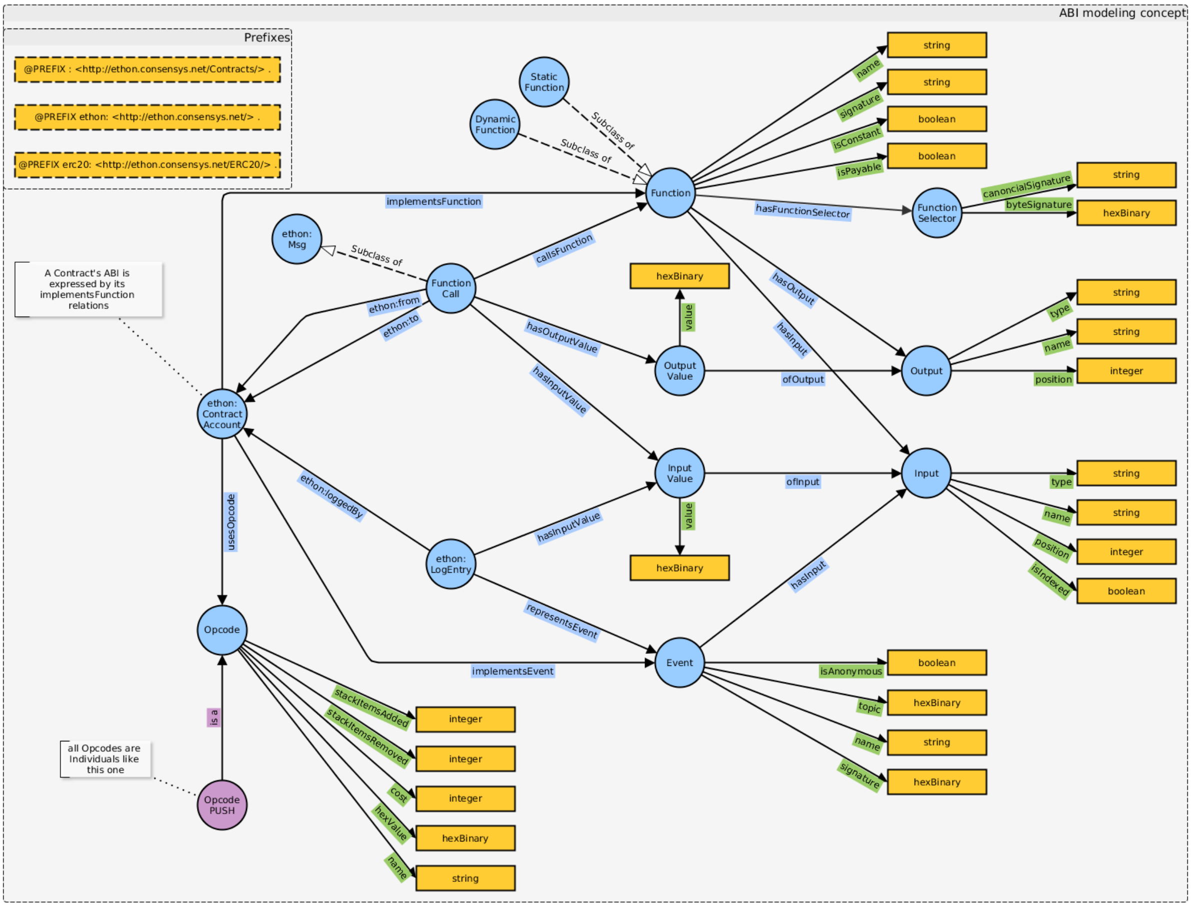


Figure 31 – EthOn Smart Contract extension[43]

6.2.2.2 *OASIS and Smart Contract Extension*

Ontology for Agents, Systems, and Integration of Services (OASIS) is a behaviour-oriented, foundational OWL 2 ontology for representing agents and their interactions. Fully transparent communication and high-level interoperability is accomplished through a request-execution protocol based on a mutual exchange of ontology fragments. The OASIS ontology that originally was focused on Internet of Agents applications, has recently been extended to allow agents to establish agreements by introducing ontological smart contracts (OSCs) [34]. OASIS

[42] https://ethon.consensys.net/Contracts

[43] https://github.com/ConsenSys/EthOn/tree/master/Contracts

adopts the concept of conditionals, OWL sentences, that are used to impose constraints on the execution of actions between agents or for verification before triggering events [35]. Conditionals are widely used outside the context of smart contracts and allow to combine constraints with behaviours to provide a higher-level description of agents and the relationships between them. In the context of smart contracts, conditionals are used to express contracts and their terms.

The concept of conditionals offers smart contracts the abstraction layer that is necessary for them to be defined at a higher level. Consequently, it is detached from the underneath layer of the specific distributed computing logic of the blockchain platform. Figure 32 depicts the core model of the ontology, showing the main classes and properties. In OASIS the main class is the SmartContract, while the smart contract instances are introduced as the class SmartContractInstance and the smart contract type as the class SmartContractType. The object-properties consistsOfSmartContractInstance and hasSmartContractType link the aforementioned instances with the instances of the SmartContract class.

This extension includes relevant smart contract concepts that need to be represented in COGITO.

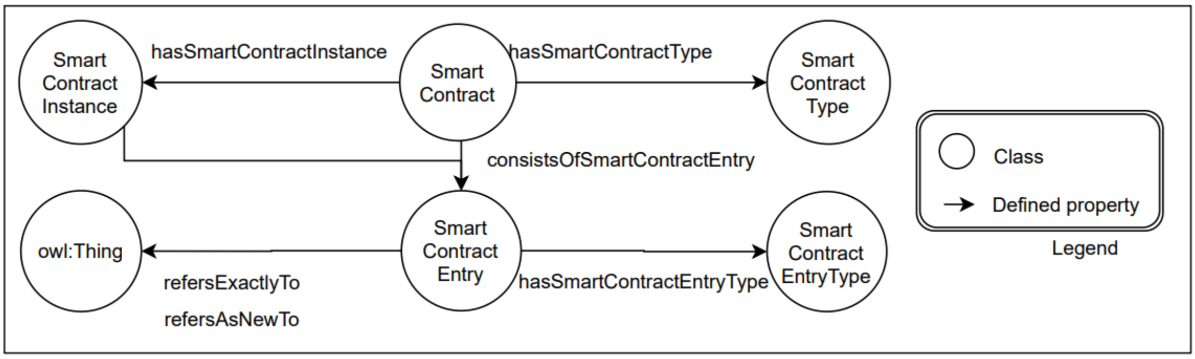


Figure 32 – OASIS Smart Contract Ontology [34]

7 Construction Safety

7.1.1 Relevant Data Models

7.1.1.1 STriDE Data Model

The spatio-temporal trajectories in dynamic environment (STriDE) data model has been developed for moving objects with non-static geometry and semantic information as time progresses [36]. The changing behaviour is represented as trajectories' time slices, which contain an identity, alphanumeric properties, a time component, and a geographical component. In Figure 33, user, trajectory and room are real world entities that may change over time, and each entity is associated with a time slice (TS) with defined start and end dates. A concept is an object with a set of attributes. In [36] those dynamic entities represent the workers, machinery and construction site workspaces based on location data from Bluetooth beacons.

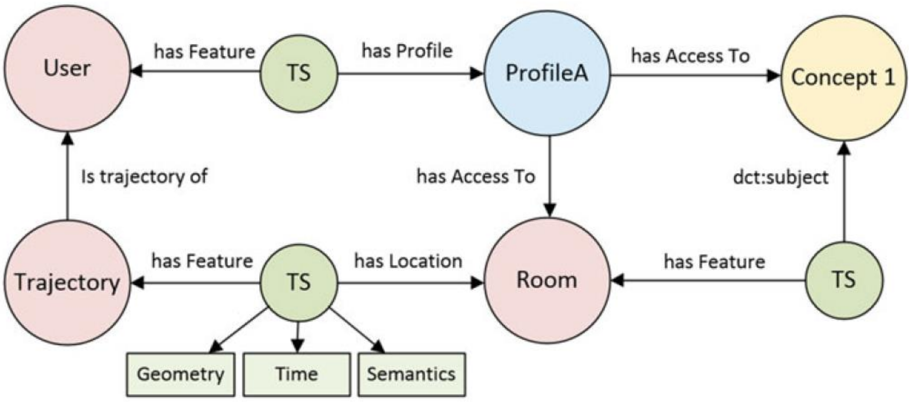


Figure 33 – STriDE (spatio-temporal trajectories in dynamic environment) data model for safety in construction [36]

The raw spatio-temporal trajectories from STriDE data model are transformed to semantic trajectories (i.e., containing semantic information) based on OpenStreetMap data files and a set of semantic rules written in the Resource Description Framework (RDF) language using the Simple Knowledge Organisation System (SKOS) vocabulary[44].

In COGITO, we are specifically interested in keeping the positions (and potentially the orientation) of humans, materials, and machines [37]. Knowledge about the trajectories of resources can have different benefits such as planning and optimising processes on the construction site but also to improve the health and safety of the crew members by actively avoiding collisions among different resources [38]. Specifically, we can utilise mobile warning systems [39] to warn workers of impending hazards, so-called close calls.

7.1.2 Relevant Ontologies

7.1.2.1 SafeConDM

SafeConDM [40] in an ontology based on previous research in ontological and logic-based approaches to Construction Safety including [41], [42], [43], [44] and [45]. Figure 34 illustrates the Construction Safety Ontology by Zhang, Boukamp & Teizer [43] extended with new (abstract) classes: spatial artefact (Spatial_Artifact) and hazard space (Hazard_Space). The authors distinguish the following three modelling layers: (1) Construction Product Model: building products and relations, such as doors, walls, storeys, slabs, and so on; (2) Construction Process Model: the construction plan including resources (equipment, materials, labour); (3) Construction Safety Model: construction safety knowledge (potential hazards, regulations, mitigating steps).

Spatial artefacts capture semantic information about regions of empty space based on construction site activities, and human perception and behaviour (movement, visibility, falling spaces, function, etc.). Similarly,

[44] https://www.w3.org/TR/skos-reference/

hazards are spatial artefacts whose existence and (geometric) definition is often a simple expression involving topological relations and Boolean operations between regions (intersection, union, offset etc.). Spatial artefacts inherit from the abstract class Product.

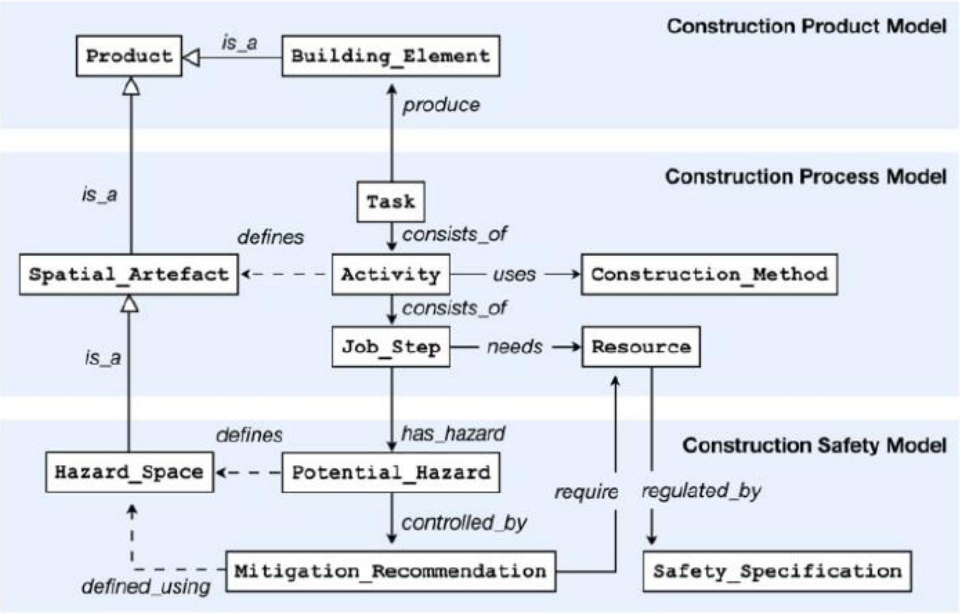


Figure 34 – SafeConDM approach [40]

In COGITO, spatial artefacts from this ontology could be used to visualise safety hazards on an updated BIM.

7.1.2.2 ETSI SAREF4WEAR

SAREF4WEAR[45] is the SAREF (see section 5.2.1.1) extension for the wearables domain. Figure 35 presents an overview of the classes and the properties included in the SAREF4WEAR extension.

saref:FeatureOfInterest describes the different actors that can be equipped with a s4wear:Wearable device. Different types of actors are foreseen: living organisms (s4wear:LivingOrganism) and software (s4wear:Software). There is also a wearer class (s4wear:Wearer) to describe those living organisms that wear some wearable.

The s4wear:LivingOrganism concept represents any living being that can be equipped with a s4wear:Wearable. The s4wear:Software concept represents a program that can be linked with a s4wear:Wearable, especially for acquiring information.

The s4wear:Wearer concept defines any saref:LivingOrganism for which the s4wear:featureIsMeasuredByDevice property subsists, i.e., the s4wear:Wearable device transmits information related to the connected saref:LivingOrganism.

The s4wear:User concept refers to a saref:FeatureOfInterest for which the s4wear:interactsWith relationship with a s4wear:Wearable individual exists.

[45] https://saref.etsi.org/saref4wear/v1.1.1/

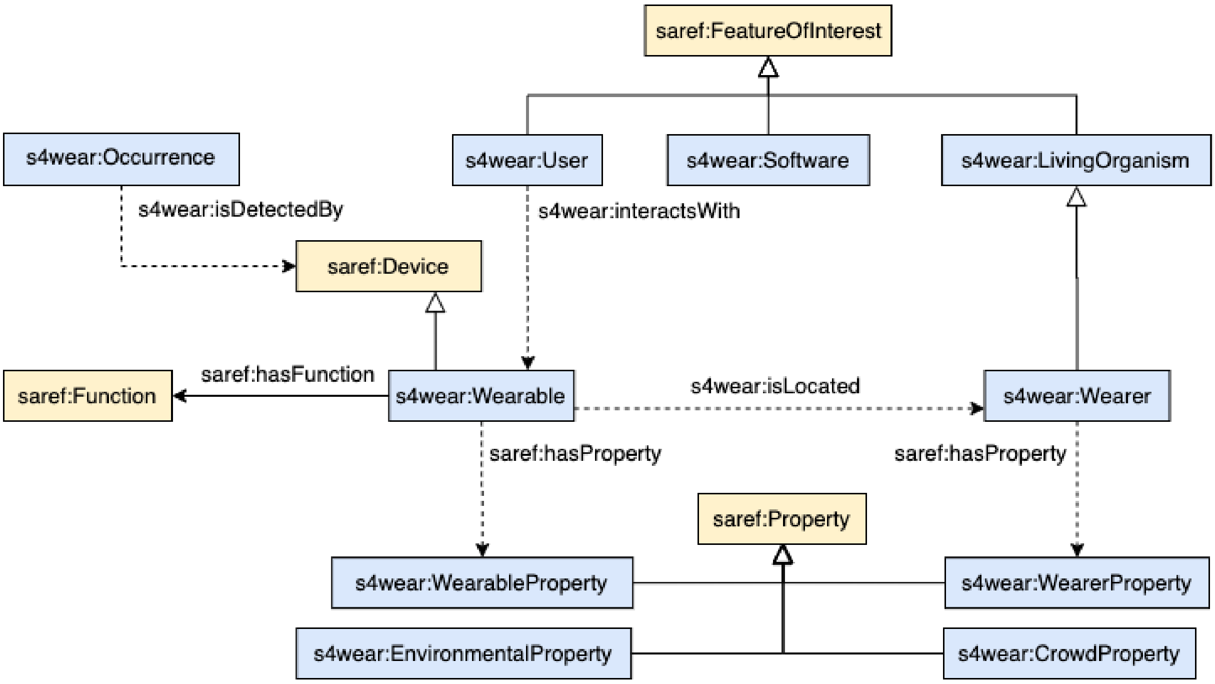


Figure 35 – SAREF4WEAR overview [46]

SAREF4WEAR defines the s4wear:Wearable abstract concept representing a saref:Device to have the capability of being wore by a s4wear:Wearer.

Besides, the capabilities of a wearable under specific conditions (ssn-system:SystemCapability), such as its precision or accuracy, can be represented using the ssn-system:hasSystemCapability property.

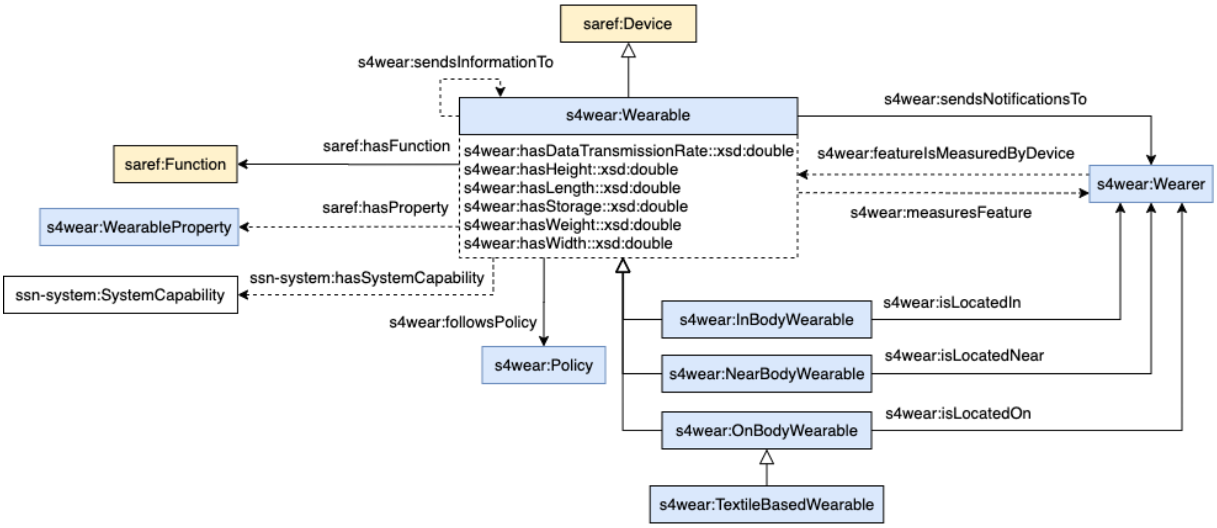

saref:Device
s4wear:sendsInformationTo
s4wear:Wearable
s4wear:sendsNotificationsTo
saref:hasFunction
saref:Function
s4wear:hasDataTransmissionRate::xsd:double
s4wear:hasHeight::xsd:double
s4wear:hasLength::xsd:double
s4wear:hasStorage::xsd:double
s4wear:hasWeight::xsd:double
s4wear:hasWidth::xsd:double
s4wear:featureIsMeasuredByDevice
s4wear:Wearer
saref:hasProperty
s4wear:WearableProperty
s4wear:measuresFeature
ssn-system:SystemCapability
ssn-system:hasSystemCapability
s4wear:followsPolicy
s4wear:Policy
s4wear:isLocatedIn
s4wear:InBodyWearable
s4wear:isLocatedNear
s4wear:NearBodyWearable
s4wear:isLocatedOn
s4wear:OnBodyWearable
s4wear:TextileBasedWearable

Figure 36 – Wearable model

The saref:Function concept specifies the functions that are considered relevant for the wearables domain. Three new concepts are defined, while other functions coming from SAREF are reused:

- saref:ActuatingFunction defines the possibility of a s4wear:Wearable device to actuate over a feature of interest;
- s4wear:CommunicatingFunction is a type of saref:EventFunction that defines the possibility of a s4wear:Wearable device to transmit data to another s4wear:Wearable device or any other saref:Device able to receive data.

- s4wear:ControllingFunction defines the possibility of a s4wear:Wearable device to control another object;
- s4wear:NavigatingFunction defines the possibility of a s4wear:Wearable device to provide navigation capabilities;
- saref:SensingFunction defines the possibility of a s4wear:Wearable device to acquire data by means of sensors integrated into a s4wear:Wearable device.

In some cases, wearables will be able to detect occurrences (s4wear:Occurrence) taking place (s4wear:takesPlaceAt) in a location that is relevant to the wearer (geosp:Feature). These occurrences can be related to the device detecting them through the s4wear:isDetectedBy property.

In the context of a smart city, more specific classes can be used from SAREF4CITY, to represent events (s4city:Event, a subclass of s4wear:Occurrence) that take place at (s4city:takesPlaceAtFacility) facilities (s4city:Facility, a subclass of geosp:Feature).

SAREF4WEAR includes a classification of the different properties that are relevant to the wearables domain. These properties are classified into wearable (s4wear:WearableProperty), wearer (s4wear:WearerProperty), crowd (s4wear:CrowdProperty), and environment (s4wear:EnvironmentProperty) ones.

The COGITO Safety component will use position data attached to workers (e.g., via wearables), so concepts from this ontology can be relevant to represent this kind of information in the project.

8 Quality Control

8.1.1 Relevant Data Models

8.1.1.1 Observations Data Model

A COGITO component for quality control is the VisualQC tool. This tool focuses on detecting typical defects in construction (cracks, bug holes etc.) and estimating their severity, by exploiting visual data acquired on site.

In the literature review the following data model was found. It considers to be relevant to our case and probably could be useful for developing a new data model for quality control in construction.

Hornsburgh et al. [47] proposed a data model for environmental and water resources data, which they called Observations Data Model (ODM). It provides a format for the storage and retrieval of point environmental observations collected by multiple sources in a relational database. Since the design identifies the entities, attributes and relations needed to represent observations in any system, it can be used in other fields as well. The model's design also includes the most important attributes from the set of Environmental Sampling, Analysis and Results Data Standards, developed by the Environmental Data Standards Council (EDSC) in 2006 [48].

The ODM Logical Data Model (see Figure 37), encodes observations (an event that results is a value describing some phenomenon) and their supporting metadata (measurement attributes, names, units, precision, accuracy, data layout and how the data was acquired). The observations in the environmental field are identified by some characteristics, such as the location that they were made (space), the date/time that they were made (time), the type of variable that was observed (variable) and many other attributes that provide additional information for the data interpretation. The geographic location of monitoring sites is specified through latitude, longitude coordinates and elevation information recorded in the Sites table. An offset is sometimes used to record the location of an observation relative to an appropriate local reference point (i.e. depth below the water surface). In some cases, such local reference is required for proper interpretation of the data. The variables can be represented in ODM range. The most fundamental attribute of an environmental variable is its name (e.g., temperature), but several other important variable attributes are recorded in ODM. All this information (such as time support, data types, samples and methods) is represented at the variable level within ODM. Raw sensor data usually contain a variety of errors caused by equipment malfunction, instrument drift etc. Data versioning and quality control are key concepts in environmental data management, where raw data streams in from in situ sensors. Therefore, each observation stored in ODM is assigned a quality control level that indicates the level of quality control to which a value has been subjected and this information is stored in the Quality Control Levels table. Furthermore, each observation stored in ODM can be attributed with an indication of the accuracy of the observation (a numeric value that quantifies the total measurement accuracy defined as the nearness of a measurement to the true or standard value). In addition, ODM offers the capability to associate observations into logical groups using the Groups and Group-Descriptions tables, which maintain association between related data values and to accompany the observations by comments that qualify how the data should be interpreted or used. Moreover, ODM provides a link for each observation in the database to the Sources table that holds information about the organization that originally collected the data while also it uses a controlled vocabulary (with a carefully selected list of words and phrases that describe units of information or data). Finally, in the ODM Series Catalog table a list of all the data series is maintained within the database giving the users the capability to search for the data they are looking for and providing them enough information to retrieve the data from the database.

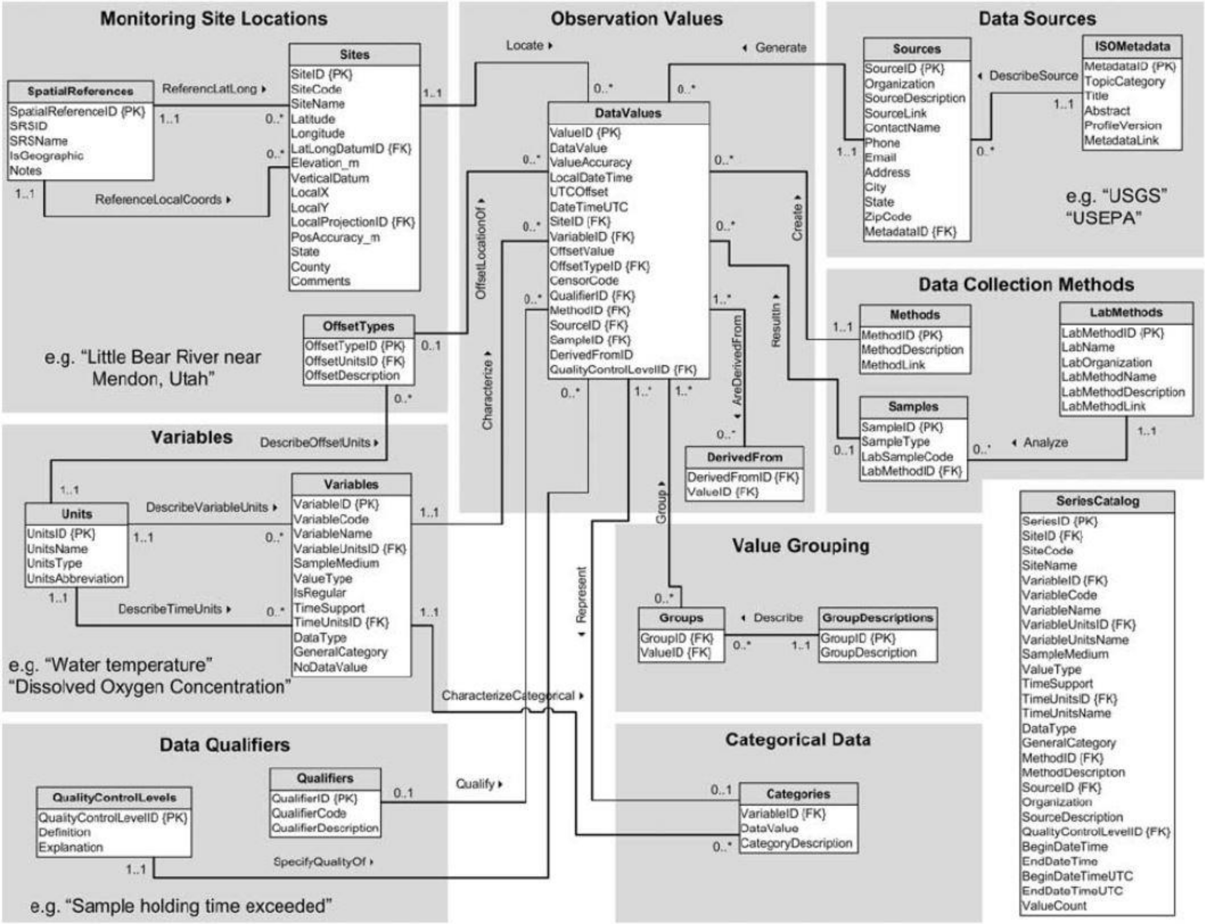


Figure 37 – Observations Data Model (ODM) logical data model [47].

8.1.2 Relevant Ontologies

8.1.2.1 BIM-based Automatic Selection Tool of Quality Control Specifications Ontology

Martinez et al. [49] proposed an extension to existing ontologies models (e.g. [50], [51]) based on OWL to represent the concepts and relationships relevant to quality control of building components. Interestingly, they developed that ontology and studied its integration as an extension to the wider ecosystem of Linked Data ontology relevant to the domain, including ifcOWL (see section 3.1.2.2). The ontology extension was developed to automatically identify regulation and standards, as well as company specifications, which are linked to the desired construction product, facilitating the identification of which criteria and KPIs each element or product must satisfy in order to pass the quality control. Figure 38 depicts the developed ontology model, which contains three main modules detailed in Figure 39: *Construction-oriented Product Ontology*, *Manufacturing-oriented Product Ontology* and *Quality Control Ontology*. The *Construction-oriented Product Ontology* module contains the building elements information, such as windows, doors, frames, studs, etc, and provides the main connection interface between the ontology and the BIM platform. The *Manufacturing-oriented Product Ontology* module contains the manufacturing operations, such as a set of construction methods, which may differ depending on the product material, required for the construction product. The *Quality Control Ontology* module is the one in charge of linking the construction quality control and assurance information, containing the set of elements and information that needs to be fulfilled to satisfy the QC.

This ontology relates strongly to the needs of the COGITO, although it may be insufficient to represent all types of specifications, especially when these refer to more than one component.

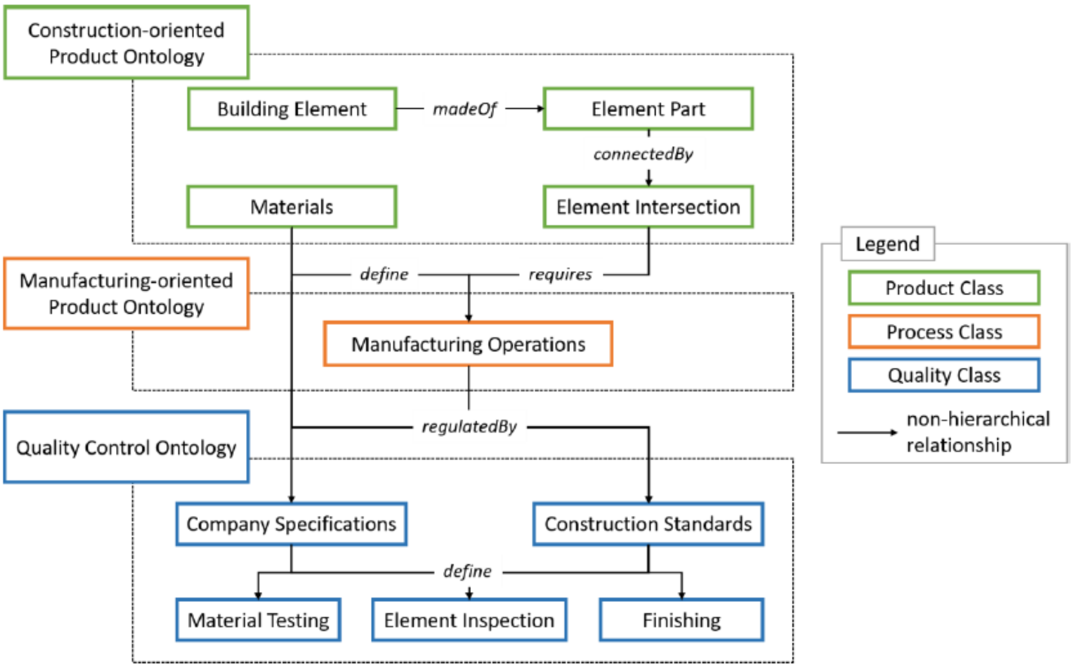
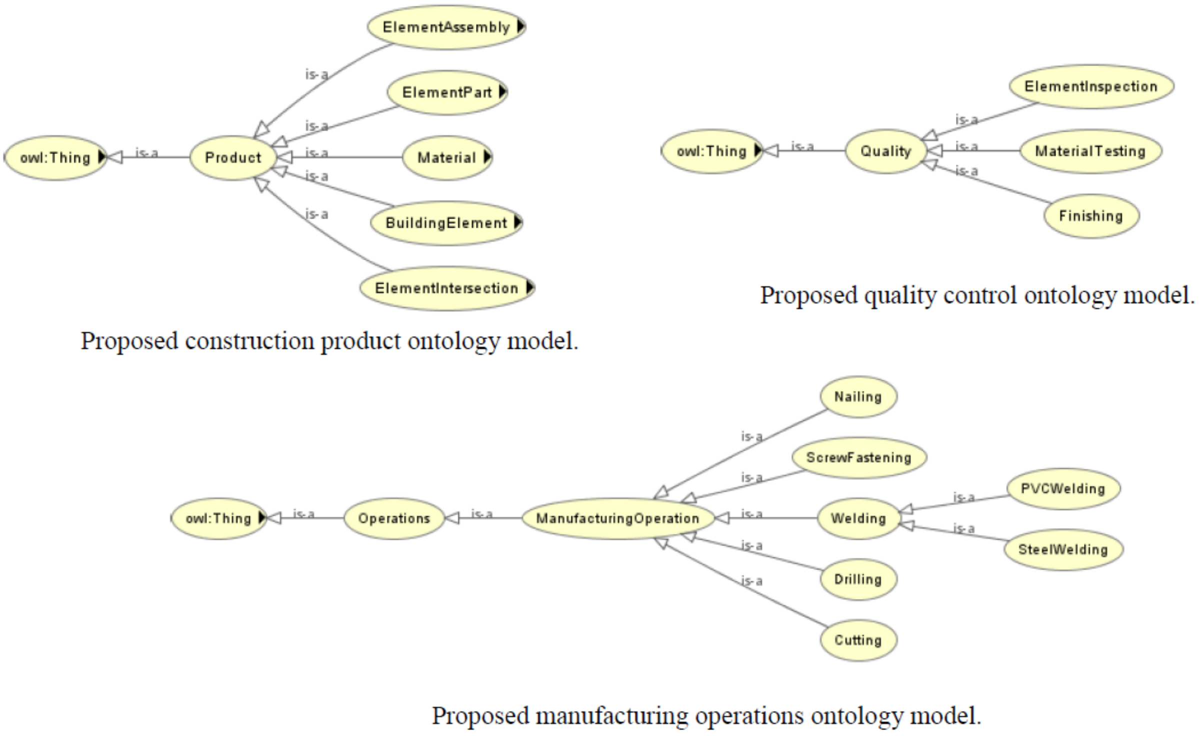


Figure 38 – Ontology model for quality control in construction manufactured products [49]



Proposed construction product ontology model.

Proposed quality control ontology model.

Proposed manufacturing operations ontology model.

Figure 39 – Detailed Ontology models [49]

8.1.2.2 CQIEOntology

CQIEOntology (Construction Quality Inspection and Evaluation) was introduced by Zhong et al. [52] to serve as a meta-model, defining the generic terms and relations related to the construction quality compliance checking, including the regulations, procedures, and evaluation criteria. The CQIEOntology (see Figure 40) is based on a main class called *Inspection-Task*¸ having all the other classes as children. The main classes are *Inspection-Object*, which refers to any entities governed by regulations and indicates what is to be inspected, *Inspection-Item-Checking-Action*, which collects the quality information/data for the inspection items, *Evaluation-Task*, *Evaluation-Criteria* and *Checking-Result*, which evaluate and verify the results from the inspection and judge whether the inspection items are compliant with the regulation constraints or not.

The CQIEOntology can be very helpful for COGITO to define a step-by-step procedure to conduct the quality control and verify that all conditions are aligned to validate the QC and finalise the construction task.

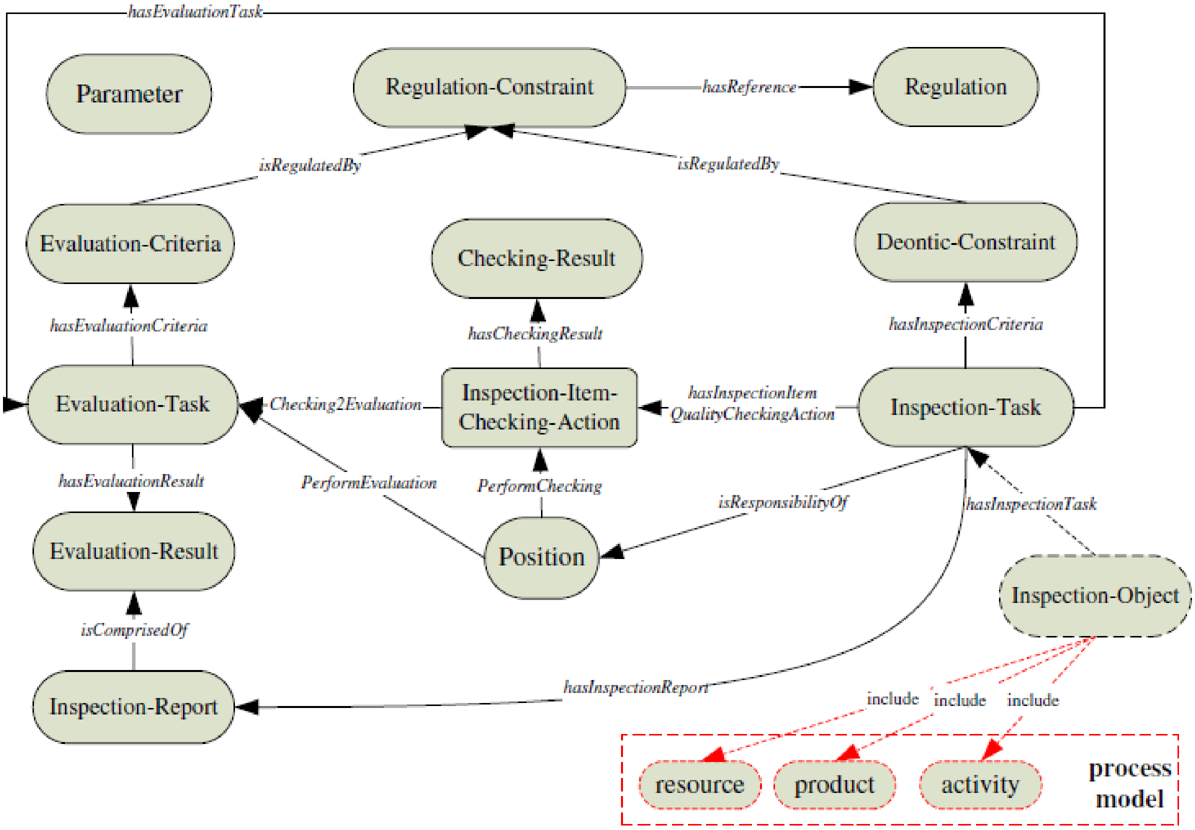


Figure 40 – CQIEOntology for construction quality compliance checking [52]

8.1.2.3 Defect Domain Ontology

C.-S. Park et al. [53] developed a defect domain ontology on the basis of the six information categories (‘General Information’, ‘Root Causes Analysis’, ‘Impact Analysis’, ‘Control Factor Analysis’, ‘Defect Description’, and ‘Media Information’) in the defect data collection template for the user to retrieve necessary defect information easily, even without having specific defect knowledge. These six categories are called as classes while their interrelationships with the defect domain are defined for the explanation of attributes of objects in the classes, such as ‘Categorize’, ‘Incur’, ‘Has’, ‘Use’, Consists of, and so on. The structure and relationships of the defect domain ontology are illustrated in this ontology (see Figure 41), different users could acquire different information (i.e. designers can search defect information by focusing on the ‘Space’ and ‘Elements’, builders can search by ‘Work Type’). This defect domain ontology is connected with a comprehensive defect data template. It would help users to identify and easily access the most relevant and critical defect information. It could also be utilized for a proactive and project-specific defect management plan as well as an education tool in areas of design and construction.

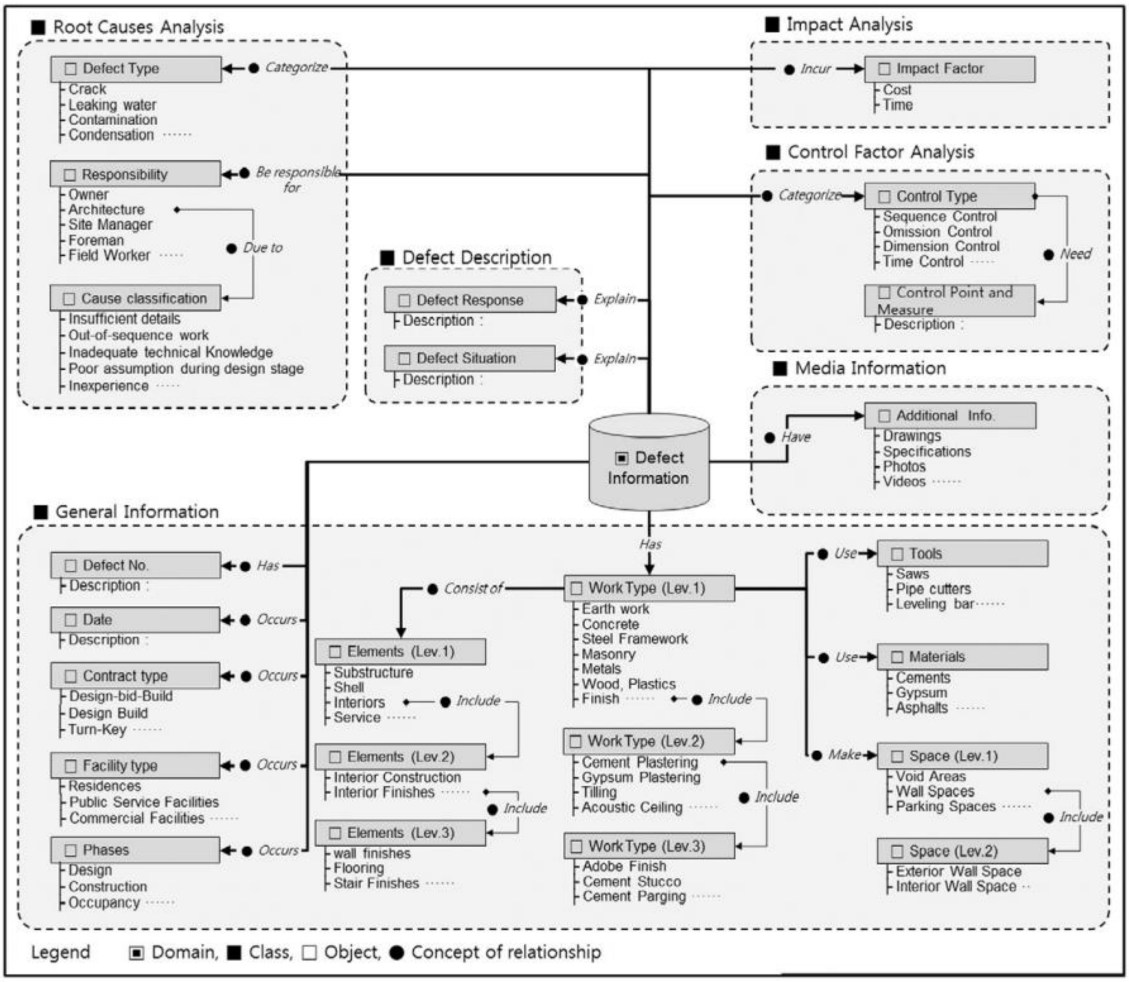


Figure 41 – Defect specific domain ontology [53].

8.1.2.4 Bridge Maintenance Ontology

BrMontology (Bridge Maintenance Ontology) emphasises on one specific type of bridge damage, i.e., crack. The ontology can automatically identify the level of the crack and evaluate the deducting point, thereby calculating the technical condition of the bridge member, component, and structure. This way, the overall bridge condition can be computed. In addition, the ontology can provide information related to bridge maintenance. It can generate and list potential reasons of different cracks, suggest maintenance of the crack and required materials for fixing, always based on features of cracks and corresponding regulations. BrMontology can also help selecting the suitable material suppliers for each case, while containing a list of material suppliers and essential information (company names, phone numbers etc.) Finally, it considers to be helpful to check information about big events and to arrange time to conduct bridge inspections and maintenances. All crucial information about big events will be listed (i.e. the location, distance, start time, end time of a marathon) and its influence on the bridge to avoiding the sharp traffic jam.

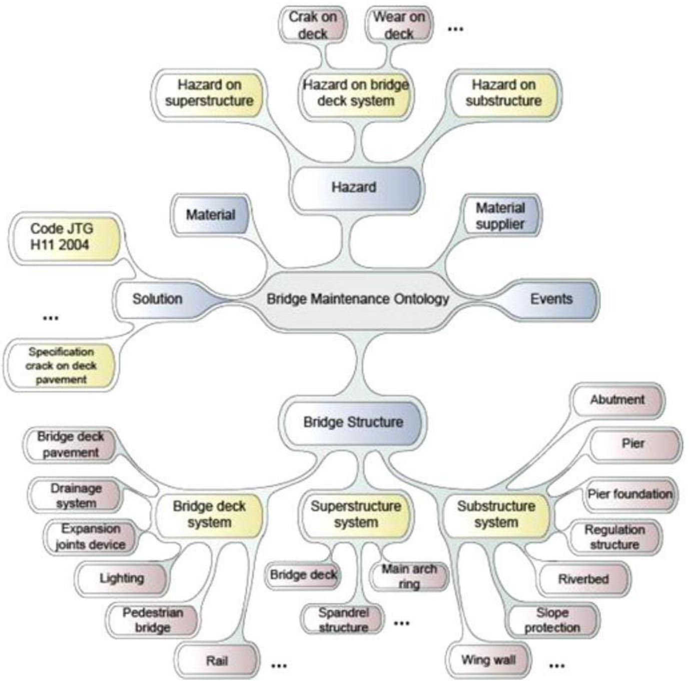


Figure 42 – Detailed classes and class hierarchy of BrMontology [54].

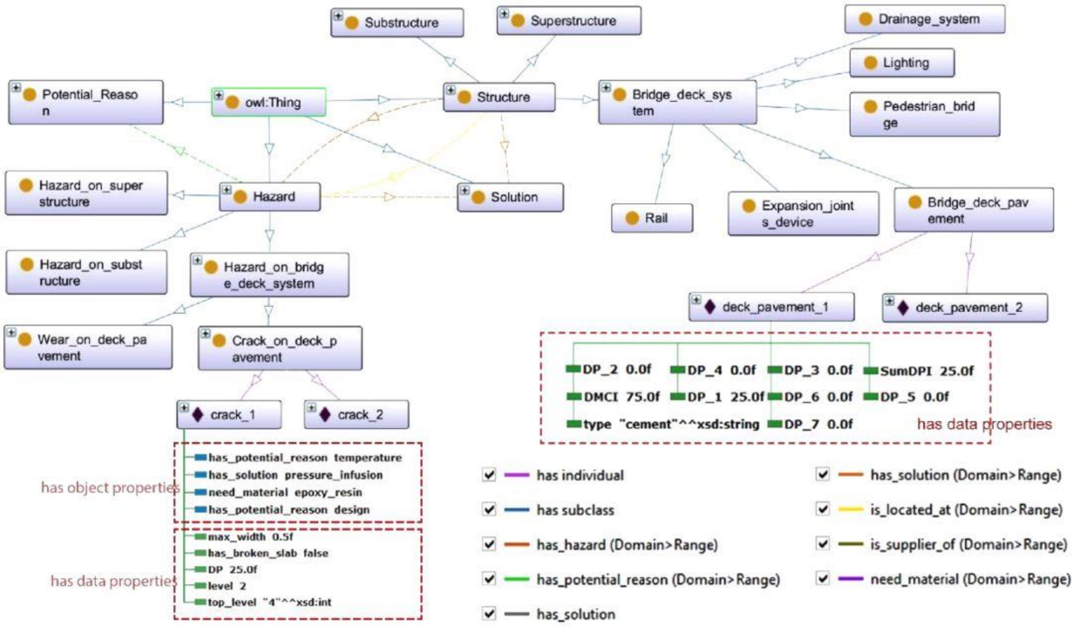


Figure 43 – A high-level overview of BrMontology [54].

9 Conclusions

This document reports the results of a survey on existing data models and ontologies related to the COGITO project. This survey is grouped into eight domains relevant to COGITO: building, process, multi-source visual data, Internet of Things (IoT), simulation, workflow management and smart contracts, health and safety, and quality control.

Table 3 amalgamates the data models and ontologies that have been described in the sections above and are relevant to the aforementioned domains. In this table, the General domain has been added to refer to those standards that cover a super set of the COGITO domains, i.e., the IFC and cityGML data models and the DICO and ifcOWL ontologies.

From this table, the gap between the COGITO ontology requirements and existing ontologies and data models that might be reused during the project, is identified. For instance, the need for new ontologies relevant to the workflow management and multi-source visual data becomes apparent, since our survey did not reveal any ontologies relevant to these domains. Furthermore, data models related to construction crew health and safety, simulation and IoT are not included in this table due to lack of existing data models targeting to these domains.

Table 3 – Relevant data models and ontologies for each COGITO domain

Domain	Data model	Ontology
General	• IFC • cityGML	• DICO ontologies • ifcOWL • BIMERR ontologies
Building	-	• SAREF4BLDG • W3C BOT ontology • Brick schema • BPO
Process	• IFC Construction, Management, Process extension • Business Process Model and Notation	• W3C Time • BBO • DICO's Process Ontology
Multi-source visual data	• ES7 • Image data model by Grosky and Stanchev • Image data model by Clouard et al.	• Ontology model of the image object features -
Internet of Things	-	• SAREF ontology • SOSA/SSN ontology • Haystack model • W3C WoT Thing Description ontology
Simulation	-	• STATO
Workflow management	• ADOxx-based Modelling Tools data models • Novitech's i3D data model	-
Smart contracts	• bcBIM data model	• EthOn • OASIS and Smart contract extension
Construction safety	-	• SAREF4WEAR
Quality control	• GeometricQC tool • Observations data model	• BIM-based Automatic Selection Tool of Quality Control Specifications Ontology • CQIEOntology • Defect domain ontology • BrMontology

This document will serve as input for the task "T3.2 – COGITO Data Model, Ontology Definition and Interoperability Design" activities, where the abovementioned data models and ontologies will be evaluated to estimate how well they cover the semantic scope of the COGITO solution. In other words, the content of these data models and ontologies will be further analysed to identify parts of them that might be reusable and to emerge the need for additional data models and ontologies that will be required to be developed from scratch towards designing, developing and delivering the COGITO ontology.

References

[1] V. Pătrăucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis and C. Haas, “State of research in automatic as-built modelling,” *Advanced Engineering Informatics,* vol. 29, no. 2, pp. 162-171, 2015.

[2] M. Poljanšek, “Building Information Modelling (BIM) standardization,” JRC Techinical Reports, 2017.

[3] L. Joblot, T. Paviot, D. Deneux and S. Lamouri, “Literature review of Building Information Modeling (BIM),” IFAC-PapersOnLine 50, 2017.

[4] E. Halmetoja, “The conditions data model supporting building information models in facility management,” *Facilities,* vol. 37, pp. 7-8, 2019.

[5] C. Eastman, P. Teichoiz and R. Sacks, BIM handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, John Wiley & Sons, 2011.

[6] SPHERE, “Digital Twin definitions for Buildings,” 2019.

[7] Q. Lu, X. Xie, J. Heaton, A. K. Parlikad and J. Schooling, “From BIM Towards Digital Twin: Strategy and Future Development for Smart Asset Management,” in *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future (SOHOMA 2019)*, 2020.

[8] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi and X. Gao, “A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends,” *Autom. Constr.,* vol. 101, 2018.

[9] W. I.Grosky, P. L.Stanchev, An Image Data Model. Advances in Visual Information Systems, Berlin Heidelberg: Springer, 2000.

[10] S. S. Leu and C. M. Chang, “Bayesian-network-based safety risk assessment for steel construction projects,” *Accident Analysis and Prevention,* vol. 54, pp. 122-133, 2013.

[11] R. A. Collazo, C. Görgen and J. Q. Smith, Chain Event Graph, Chapman and Hall, 2018.

[12] M. Crosby, Nachiappan, P. Pattanayak, S. Verma and V. Kalyanaraman, “BlockChain Technology: Beyond Bitcoin,” Sutardja Center for Entrepreneurship & Technology Technical Report. University of California, Berkeley., 2015.

[13] Eurostat, “Fatal and non-fatal accidents at work by NACE Section, EU-27,,” 2018.

[14] S. Zhanga, F. Boukampb and J. Teizer, “Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA),” *Automation in Construction,* pp. 29-41, 2015.

[15] EN13670:2009, *Execution of concrete structures,* 2009.

[16] “ISO 9001 Quality Management,” [Online]. Available: https://www.iso.org/iso-9001-quality-management.html.

[17] M.-O. Löwner, J. Benner, G. Gröger and K.-H. Häfele, “New concepts for structuring 3D city models--an extended level of detail concept for CityGML buildings,” in *International Conference on Computational Science and Its Applications*, 2013.

[18] M. El-Mekawy, A. Östman and K. Shahzad, “Towards interoperating CityGML and IFC building models: a unified model based approach,” in *Advances in 3D Geo-Information Sciences*, Springer, 2011, pp. 73--93.

[19] “ETSI TS 103 410-3 V1.1.2: SmartM2M; Extension to SAREF; Part 3: Building Domain,” 2020.

[20] M. H. Rasmussen, P. Pauwels, C. Hviid and J. Karlshøj, “Proposing a Central AEC Ontology That Allows for Domain Specific Extensions,” in *Joint Conference on Computing in Construction*, 2017.

[21] M. H. Rasmussen, P. Pauwels, M. Lefrançois, G. Schneider, C. Hviid and J. Karlshøj, “Recent changes in the Building Topology Ontology,” in *Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC 2017)*, 2017.

[22] A. Wagner and U. Rüppel, “BPO: The Building Product Ontology for Assembled Products,” in *Proceedings of the 7th Linked Data in Architecture and Construction workshop (LDAC 2019)*, 2019.

[23] A. Annane, N. Aussenac-Gilles and M. Kamel, “BBO: BPMN 2.0 Based Ontology for Business Process Representation,” in *20th European Conference on Knowledge Management (ECKM'19)*, Lisbon, 2019.

[24] D. Huber, “The ASTM E57 file format for 3D imaging data exchange,” in *Proc. SPIE 7864*, 2011.

[25] “ASTM E2807 Standard,” [Online]. Available: https://www.astm.org/Standards/E2807.htm.

[26] V. Gudivada, V. Raghavan, K. Vanapipat, “A Unified Approach to Data Modeling and Retrieval for a Class of Image Database Applications,” *IEEE Transactions on Data and Knowledge Engineering,* 1994.

[27] A. Gupta, T. Weymouth, R. Jain,, “Semantic Queries with Pictures: The VIMSYS Model,” in *Proceedings of the 17th Conference on Very Large Databases*, Palo Alto, California, 1991.

[28] M. Mechkour, EMIR2. An Extended Model for Image Representation and Retrieval, Berlin: Springer -Verlag, 1995.

[29] R.Clouard, A. Renouf, M. Revenu, “An Ontology-Based Model for Representing Image Processing Objectives,” *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 24, no. 8, pp. 1181-1208, 2010.

[30] H. Gu, H. Li, L. Yan, Z. Liu, T. Blaschke, U. Soergel, “An Object-Based Semantic Classification Method for High Resolution Remote Sensing Imagery Using Ontology,” *Remote Sensing.,* vol. 9, no. 329, 2019.

[31] “ETSI TS 103 264 – v3.1.1. SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping.,” 2020.

[32] A., Haller; K., Janowicz; S., Cox; M., Lefrançois; K., Taylor; D., Le Phuoc; J., Lieberman; R., García-Castro; R., Atkinson; C., Stadler, “The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Samplings and Actuation,” *Semantic Web - Interoperability, Usability, Applicability. IOS Press. ISSN: 1570-0844,* vol. 10, no. 1, pp. 9-32, January 2019.

[33] R. Zheng, J. Jiang, X. Hao, W. Ren and F. Xiong, “bcBIM: A Blockchain-Based Big Data Model for BIM Modification Audit and Provenance in Mobile Cloud,” *Hindawi Mathematical Problems in Engineering,* vol. 2019, p. 13, 2019.

[34] D. Cantone, C. F. Longo, M. Nicolosi-Asmundo, D. F. Santamaria and C. Santoro, “Ontological Smart Contracts in OASIS: Ontology for Agents, Systems, and Integration of Services,” in *14th International Symposium on Intelligent Distributed Computing*, Scilla, Reggio Calabria, Italy, 2020.

[35] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean, “SWRL: A Semantic Web rule language combining OWL and RuleML,” *W3C Subm,* vol. 21, 2004.

[36] M. Arslan, C. Cruz and D. Ginhac, “Identifying Intrusions in Dynamic Environments Using Semantic Trajectories and BIM for Worker Safety,” in *Fourth International Congress on Information and Communication Technology, Advances in Intelligent Systems and Computing*, 2020.

[37] J. Zhao, O. Seppänen, A. Peltokorpi and B. Badi, “Real-time resource tracking for analyzing value-adding time in construction,” pp. 52-65, 2019.

[38] K. M. Rashid and A. H. Behzadan, “Risk behavior-based trajectory prediction for construction site safety monitoring.,” *Journal of construction engineering and management,* vol. 144, no. 2, 2018.

[39] J. Teizer, B. S. Allread, C. E. Fullerton and J. Hin, “Autonomous pro-active real-time construction worker and equipment operatorproximity safety alert system,” *Automation in Construction,* pp. 630-640, 210.

[40] C. Schultz, B. Li and J. Teizer, “Towards a Unifying Domain Model of Construction Safety: SafeConDM,” in *EG-ICE 2020 Proceedings : Workshop on Intelligent Computing in Engineering*, 2020.

[41] S. Zhang, J. Teizer, J. K. Lee, C. M. Eastman and M. Venugopal, “Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules,” *Automation in,* vol. 29, pp. 183-195, 2013.

[42] S. Zhang, K. Sulankivi, M. Kiviniemi, I. Romo, C. M. Eastman and J. Teizer, “BIM-based fall hazard identification and prevention in construction safety planning,” *Safety science,* vol. 72, pp. 31-45, 2015.

[43] S. Zhang, F. Boukamp and J. Teizer, “Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA),” *Automation in Construction,* pp. 29-41, 2015.

[44] J. Teizer, “Right-time vs. Real-time Pro-active Construction Safety and Health System Architecture,” *Construction Innovation: Information, Process, Management, Emerald,* vol. 16, no. 3, pp. 253-280, 2016.

[45] K. Schwabe, J. Teizer and M. König, “Applying rule-based model-checking to construction site layout planning tasks,” *Automation in Construction,* vol. 97, pp. 205-219, 2019.

[46] “ETSI TS 103 410-9 V1.1.1: SmartM2M; Extension to SAREF; Part 9: Wearables Domain,” 2020.

[47] J. Horsburgh, D. Tarboron, D. Maidment and I. Zaslavsky, “A relational model for environmental and water resources data,” *Water Resources Research,* vol. 44, no. W05406, 2008.

[48] Environmental Data Standards Council, *Environmental sampling,analysis, and results data standards: Overview of component data standards, Stand. EX000001.1, Environ.,* Data Stand. Counc., U. S. Environ Prot.Agency, Washington,D.C., 2006.

[49] P. Martinez, R. Ahmad and M. Al-Hussein, “Automatic Selection Tool of Quality Control Specifications for Offsite Construction Manufacturing Products: A BIM-based Ontology Model Approach,” in *Modular and Offsite Construction Summit*, Banff, 2019.

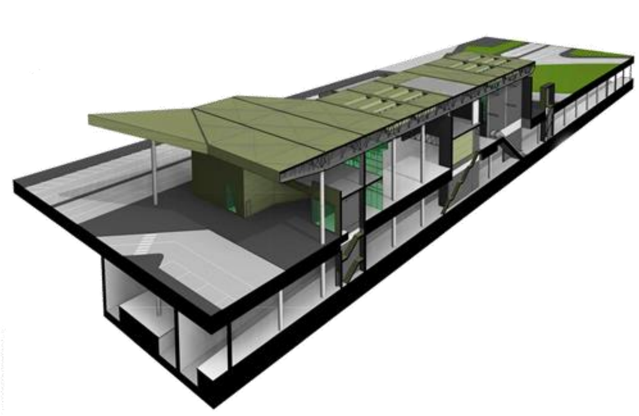
[50] S. Lemaignan, A. Sladat, J.-Y. Dantan and A. Semenenko, “MASON: A proposal for an ontology of manufacturing domain,” in *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06*, 2006.

[51] H. Liu, M. Lu and M. Al-Hussein, "Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry," *Advanced Engineering Informatics,* vol. 30, no. 2, pp. 190-207, 2016.

[52] B. Zhong, L. Ding, H. Luo, Y. Zhou, Y. Hu and H. Hu, "Ontology-based semantic modeling of regulation constraint for automated," *Automation in Construction,* no. 28, pp. 58-70, 2012.

[53] C.-S. Park, D.-Y. Lee, O.-S. Kwon, X. Wang, "A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template," *Automation in Construction,,* vol. 33, pp. 61-71, 2013.

[54] G.Ren, R. Ding, H. Li, "Building an ontological knowledgebase for bridge maintenance," *Advances in Engineering Software,* vol. 130, pp. 24-40, 2019.



COGITO

CONSTRUCTION PHASE
DIGITAL TWIN MODEL

cogito-project.eu

HYPERTECH energy labs

UCL

AARHUS UNIVERSITET

THE UNIVERSITY of EDINBURGH

www.boc-group.com

iti

POLITÉCNICA

QUE TECHNOLOGIES

Novitech New Information Technologies

ASM

ferrovial construction

ΟΛΥΜΠΙΑ ΟΔΟΣ

RHOMBERG SERSA RAIL GROUP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310