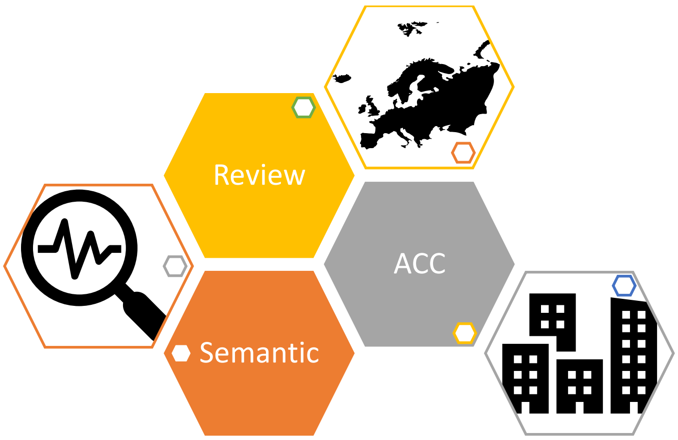
Existing ontologies, standards, and data models in the building data domain relevant to compliance checking

Aug 29, 2023



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101056973.

Funded by the European Union

UK Participants in Horizon Europe Project [ACCORD] are supported by UKRI grant numbers [10040207] (Cardiff University), [10038999] (Birmingham City University and [10049977] (Building Smart International).

Innovate UK

Project Title	ACCORD - Automated Compliance Checks for Construction, Renovation or Demolition Works, grant agreement No: 101056973.
Deliverable:	D2.1 Technical Report: Existing Models
Type:	R — Document, report
Dissemination level:	Public
Work package:	WP2: Semantisation of regulation and open format for machine-readable rules
Lead Beneficiary:	IMT
Deliverable leader:	IMT
Contributing partners:	CU, BCU, FhG, ONTO, SOL, FUI, FUNITEC, JU
Due date:	2023-08-31
Date:	2023-08-29
Status – version, date:	V1.0 – 2023-08-29
Authors:	Thamer Mecharnia (IMT)
	Maxime Lefrançois (IMT)
	Antoine Zimmermann (IMT)
	Edlira Vakaj (BCU)
	Amna Dridi (BCU)
	Hansi Hettiarachchi (BCU)
	Vladimir Alexiev (ONTO)
	Nataliya Keberle (ONTO)
	He Tan (JU)
	Francesca Noardo (OGS)
	Rick Makkinga (FUI)
	Franco Cheung (BCU)
Reviewer 1:	Gonçal Costa Jutglar (FUNITEC)
Reviewer 2:	Thomas Beach (CU)

DOCUMENT HISTORY

Version	Date	% Complete	Comments	Main Authors (organisation)
0.1	07/11/2022	5	Initial Deliverable Structure and Table of Contents	Maxime Lefrançois (IMT)
0.2	13/02/2023	27	Sections 5 and 6 ready for 1 month period review by T2.1 participants	Thamer Mecharnia (IMT), Maxime Lefrançois (IMT), Antoine Zimmermann (IMT)
0.3	09/03/2023	49	Section 3 and 4 ready for 1 month period review by T2.1 participants	Thamer Mecharnia (IMT), Maxime Lefrançois (IMT), Antoine Zimmermann (IMT)
0.4	05/04/2023	72	Section 2 ready for 1 month period review by T2.1 participants	Thamer Mecharnia (IMT), Maxime Lefrançois (IMT), Antoine Zimmermann (IMT)
0.5	24/05/2023	95	Take into account feedback from T2.1 participants review	Thamer Mecharnia (IMT), Maxime Lefrançois (IMT), Antoine Zimmermann (IMT)
0.6	27/07/2023	96	Internal review	Gonçal Costa Jutglar (FUNITEC), Thomas Beach (CU)
0.7	28/07/2023	98	Minor formatting edition	Thamer Mecharnia (IMT), Maxime Lefrançois (IMT)
0.8	25/08/2023	99	Proof reading	Franco Cheung (BCU)
0.9	29/08/2023	100	Published Version	Maxime Lefrançois (IMT)

Statement of originality:

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

Executive summary

This deliverable presents the results of Task 2.1 (Technical Review of Existing Standards) of the ACCORD project.

The ACCORD project employs a semantic approach for validating building permits, eliminating the need for costly centralized systems that are challenging to establish and maintain. The primary aim of the ACCORD project is to digitize permit and compliance procedures to improve the productivity and quality of design and construction processes and facilitate the creation of an environmentally sustainable built environment.

This deliverable will review the existing ontologies, standards, and data models in the Architecture, Engineering, and Construction (AEC) domain and how they can be reused for the purpose of the automatic compliance check. More specifically, this deliverable will:

1. Evaluate the AEC domain-related ontologies and propose suggestions on how they can be employed for the development of the Architecture Engineering and Construction Compliance Checking and Permitting Ontology (AEC3PO).
2. Conduct a review of query languages associated with the AEC domain and the semantic web.
3. Compare the rule languages developed or used in AEC projects.
4. Review the standards that may be relevant to different areas in the ACCORD project.
5. Compare the existing reasoners that could be useful to building permitting automatic compliance checking.

All the references used in this deliverable are gathered in the open Zotero library for the project: https://www.zotero.org/groups/3007408/semantic_bim/library

In the AEC industry, several standards and recommendations aim to achieve different levels of data interoperability in systems. This deliverable concentrates on data-related standards such as those that provide syntactic rules and semantics to represent data in a standardized way. Policy and regulatory standards are out of the scope of this deliverable and are addressed in deliverable D1.1 “Landscape Review Report”.

The outcomes of this deliverable will serve as a reference for other tasks within the project, which will determine the preferred rule language, which ontologies can be reused, aligned, or serve as inspiration for the creation of the AEC3PO to be developed in Task 2.2 of WP2. Furthermore, the standards that will be presented in this deliverable can be employed in various aspects of the ACCORD project.

This groundwork will facilitate the development of the AEC3PO ontology as well as the design and implementation of the Rule Formalisation Tool.

Publishable summary

This deliverable presents the results of Task 2.1 (Technical Review of Existing Standards) of the ACCORD project.

The aim of the ACCORD project is to digitize the building permitting and compliance procedures to improve the quality and productivity of design and construction processes and support the development of a sustainable built environment. This is achieved through the adoption of a semantic approach where individual tools are treated as microservices, eliminating the requirement for costly centralized systems that are hard to establish and manage.

This deliverable will review the existing ontologies, standards and data models in Architecture, Engineering, and Construction (AEC) domain and how they can be reused for the purpose of the automatic compliance check. More specifically, this deliverable will:

1. Evaluate the AEC domain-related ontologies and suggestions on how they can be employed for the Architecture Engineering and Construction Compliance Checking and Permitting Ontology (AEC3PO).
2. Conduct a review of query languages.
3. Compare the rule languages for better understanding of which rule languages are the most effective in terms of expressivity for representing building regulations.
4. Review the standards that may be relevant to different areas in ACCORD project.
5. Compare the existing reasoners.

Contents

List of Figures

List of Tables

Acronyms

ACC	Automated Compliance Checking
ACCORD	Automated Compliance Checks for Construction, Renovation or Demolition Works
ADMS	Asset Description Metadata Schema
AEC	Architecture, Engineering, and Construction
AEC3PO	Architecture Engineering and Construction Compliance Checking and Permitting Ontology
AI	Artificial Intelligence
API	Application Programming Interface
ASTM	American Society for Testing and Materials
BAO	Building Assessment Ontology
BCAO	Building Circularity Assessment Ontology
BCDI	BIM-CITYGML Data Integration
BCF	BIM Collaboration Format
BEO	Building Element Ontology
BERA	Building Environment Rule and Analysis Language
BFO	Basic Formal Ontology
BIM	Building Information Modeling
BIMERR	BIM-based holistic tools for Energy-driven Renovation of existing Residences
BIMQL	BIM Query Language
BIMRL	BIM Rule Language
BIRL	Dutch Building Information Council
BOT	Building Topology Ontology
BPO	Building Product Ontology
CAD	Computer-Aided Design
CB-NL	Concept Library for the Built Environment
CCIC	Construction Classification International Collaboration
CDE	Common Data Environments
CEN	European Committee for Standardization
COINS	Construction Industry Solutions
CoMOn	Compliance Management Ontology
CPR	Construction Products Regulation
CQIE	Construction Quality Inspection and Evaluation
CWM	Closed World Machine
DBO	Digital Buildings ontology
DCAT	Data Catalog Vocabulary

DER	Distributed Energy Resources
DFS	Design for Safety
DiCon	Digital Construction Ontologies
DIN	German Institute for Standardization
DL	Description Logic
DOCK	Domain Ontology for Construction Knowledge
DR	Demand Response
ELI	European Legislation Identifier
ETIM	European Technical Information Model
EYE	Euler Yap Engine
FAIR	Findable, Accessible, Interoperable, Reusable
FOAF	Friend of a friend
FOG	File Ontology for Geometry formats
FOL	First Order Logic
FSGIM	Facility Smart Grid Information Model
GIS	Geographic Information System
GML	Geography Markup Language
GS	Global Standards
IC-PRO	Infrastructure and Construction PROcess
IDDO	The Interconnected Data Dictionary Ontology
IEC	International Electrotechnical Commission
IFC	Industry Foundation Classes
ifcOWL	IFC in Web Ontology Language
IFC-PROPS	IFC Properties
ifcWOD	IFC Web of Data ontology
ISO	International Organization for Standardization
Jess	Java Expert System Shell
JRC	Joint Research Centre
JSON	JavaScript Object Notation
KBIM	Korea BIM
KBVL	KBim Visual Language
KPI	Key Performance Indicator
LADM	Land Administration Domain Model
LandInfra	Land and Infrastructure Conceptual Model
LBD	Linked Building Data
LinkML	Linked Data Modeling Language
LKIF	Legal Knowledge Interchange Format

MEP	Distribution Element Ontology
N3Logic	Notation 3 Logic
NDBC	Numeric Data of Building Circulation
NEN	Dutch Standardization Organization
NIBS	National Institute of Building Sciences
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NRC	National Research Council of Canada
NSIT	National Institute of Standards and Technology
NTNU	Norwegian University of Science and Technology
OASIS	Organization for the Advancement of Structured Information Standards
oBIX	Open Building Information Exchange
OBPA	Ontology for building permit authorities
OGC	Open Geospatial Consortium
OMG	Ontology for Managing Geometry
OOPS!	OntOlogy Pitfall Scanner!
OpenADR	Open Advanced Demand Response
OPM	Ontology for Property Management
ORG	Organization Ontology
OWL	Web Ontology Language
POE	Post-Occupancy Evaluation Ontology
Prolog	PROgramming in LOGic
QL4BIM	Query Language for BIM
QUDT	Quantities, Units, Dimensions and Data Types
RASE	Requirement, Applicability, Selection, and Exceptions
RDF	Resource Description Framework
RDFS	RDF Schema
REC	RealEstateCore
RIBA	UK-based Royal Institute of British Architects
RIF	Rule Interchange Format
RKQL	Regulatory Knowledge Query Language
RuleML	Rule Markup Language
SAREF	Smart Applications REFerence ontology
SEAS	Smart Energy Aware Systems
SHACL	SHApes Constraint Language
ShEx	Shape Expressions
SKOS	Simple Knowledge Organization System

SOSA	Sensor, Observation, Sample, and Actuator
SPARQL	SPARQL Protocol and RDF Query Language
SPIN	SPARQL Inferencing Notation
SQL	Structured Query Language
SRO	Safety Regulation Ontology
SSN	Semantic Sensor Network
SWRL	Semantic Web Rule Language
TC	Technical Committee
TSV	Tabulation-Separated Values
UML	Unified Modeling Language
UNICLASS	Unified Construction Classification
VCCL	Visual Compliance Checking Language
W3C	World Wide Web Consortium
WP	Work Packages
XML	Extensible Markup Language
XSD	XML Schema Definition Language

1. Introduction

D2.1 deliverable is a technical report concerning ontologies, standards, and data models within the building data domain that are relevant to compliance checking. It presents the results of Task 2.1 (Technical Review of Existing Standards) conducted as part of the ACCORD project (Automated Compliance Checks for Construction, Renovation or Demolition Works).

1.1 The ACCORD Project

The ACCORD project utilizes a semantic methodology for the building permitting validation process, thus obviating the need for expensive centralized systems that are difficult to create and maintain. The fundamental objective of ACCORD is to digitize permitting and compliance procedures with the intention of enhancing the productivity and quality of design and construction processes, as well as facilitating the construction of an ecologically sustainable built environment.

1.2 Work Package 2

The ACCORD project comprises seven Work Packages (WP), with D2.1 serving as the initial output for work package 2 titled "Semantisation of regulation and open format for machine-readable rules". Within WP2, there will be the creation of semantic models, Artificial Intelligence models, and a tool for formalizing rules. This involves constructing an ontology encompassing laws, regulations, and administrative processes related to building design, construction, and operation. Additionally, WP2 aims to establish an open format for representing rules in a platform-neutral manner, which will be facilitated by the aforementioned rule formalization tool.

The second work package (WP2) consists of six tasks:

1. **Task 2.1: Technical Review of Existing Standards**: This task involves evaluating existing ontologies, standards, and data models within the domain to determine their suitability for automated compliance checks. It includes analysing potential alignments between existing ontologies/data models and identifying any representation gaps that may need to be addressed by tasks 2.2 and 2.3.

2. **Task 2.2: Development of the Architecture Engineering and Construction Compliance Checking and Permitting Ontology (AEC3PO)**: Drawing on literature, expert interviews, task 2.1, and work package 1 titled “Requirements for digitalising permitting and compliance processes”, this task aims to create a conceptual ontological model of building compliance requirements. The AEC3PO ontology is designed to be flexible and not limited to specific regional or legal systems. It aligns with established standard ontologies for representing generic concepts (such as time, processes, and document metadata) and specific aspects of the building domain (such as building topology and construction projects). Specific laws, regulations, processes, and documentation will be implemented based on this generic model.

3. **Task 2.3: Machine-executable**: This task involves analysing existing methodologies for producing machine-executable rules and regulations. Insights will be drawn from academic literature and existing software systems, including examples from the built environment and other domains. Based on this analysis, the task will define the ACCORD methodology for making rules and regulations (semantically defined in task 2.2) machine-executable. Existing

approaches will be selected, adapted, and integrated with the semantics defined in task 2.2, with a focus on avoiding the development of new technologies. This task will also generate a ruleset database that provides semantic rules for compliance checking, facilitated by task 2.5, to support the checking processes of work packages 4 “Solutions development” and 5 “Prototype solutions demonstrations” as a proof of concept and input.

4. **Task 2.4: Artificial Intelligence for Natural Language Processing of Building**: This task involves developing a suite of Artificial Intelligence (AI) models and algorithms to extract, generate, formulate, and semantize rules from text. Deep learning Natural Language Processing (NLP) techniques will be applied to building codes and regulations to extract rule entities/axioms in the UK. Additional algorithms will be developed to incorporate the NLP output into the rule format defined in task 2.3 and semantized using the ontologies and data models from tasks 2.2 and 2.1. This task will also focus on semantizing existing digitized/automated rules.

5. **Task 2.5: Design and Implementation of Rule Formalization Tool**: This task involves designing and implementing a rule formalization tool that utilizes components developed in other tasks of WP2. The tool aims to assist rule creators in the process of formalizing regulations and information requirements into a machine-processable rule format. The first release of the tool will incorporate the outputs of tasks 2.1, 2.2, and 2.3, while the second release will include NLP processing from task 2.4. This task will also define the methodology for transforming text into a machine-interpretable rule format.

6. **Task 2.6: Technical Guidelines**: This task focuses on developing documentation for the artifacts produced within WP2. It includes creating documentation for the AEC3PO ontology (including textual descriptions and visualizations), documenting the rule formalization tool and its ability to automatically generate documentation from the artifacts, providing example instantiations of the ontology and rules for various real-world scenarios and guideline documents.

The outcomes of the study conducted in this deliverable will serve as a reference for other tasks within the project, which will determine the preferred rule language, which ontologies can be reused, aligned, or serve as inspiration for the creation of AEC3PO ontology. Furthermore, the standards presented in this deliverable can be employed in various aspects of the ACCORD project, including the syntactic representation of data, Application Programming Interfaces (APIs), and so on.

1.3 Objectives

This deliverable document how some of the ACCORD project tasks, especially those developed in WP2, can benefit from the existing standards, ontologies, query languages, rule languages, and reasoners. Every one of those elements has a dedicated section to study and analyse them.

1. **To review** Ontologies for

   The main focus of this objective is to assemble a compilation of ontologies that can serve as a source of inspiration for the AEC3PO ontology. Additionally, this objective involves conducting a thorough evaluation of the existing ontologies and providing recommendations on how they can be effectively utilized for various aspects of AEC3PO, such as building modelling and regulation modelling, and for identifying research gaps,

2. **To review Query Languages**

The objective comprises a compilation of existing query languages and illustrate their respective functionalities based on the type of data being queried, including geospatial data, and Resource Description Framework (RDF) data, etc.

3. **To review Rule Languages for ACC (ACC)**

The ACC process requires that building regulations must be converted into machine-readable rules to verify the compliance of building permitting. Therefore, we are focusing on two primary objectives: (1) showcasing the existing rules classification that exist in the current state of the art, and (2) presenting a roster of the rule languages that have been previously utilized in building inspections or have proven useful for this task. In addition, this objective also involves comparing these rules to provide a definitive understanding of which rule language is the most effective in representing regulations.

4. **To review Reasoners and Rule Engines**

The objective is delivering an analytical roster of reasoners, which are compared based on the supported Description Logic (DL) and the nature of the reasoning task (whether it involves inference, data validation, and/or query answering).

5. **To review common Standards and Recommendations**

This objective has two main goals. Firstly, it aims to collect recommendations and standards that can serve as a source of inspiration for ACCORD tasks, such as syntaxes and terminologies. Secondly, it seeks to provide an analysis of items that can influence the development of the AEC3PO ontology, while also reviewing other standards that may have relevance to different areas within the ACCORD project.

2. Ontologies for Architecture, Engineering, and Construction (AEC)

The purpose of developing the AEC3PO ontology is to encompass both general concepts (such as time, processes, and file/document metadata) and domain-specific aspects related to the building industry (such as building topology and construction projects). The primary goal of AEC3PO is to facilitate automated rule checking within permitting and compliance processes. The creation of the AEC3PO ontology can find inspiration from existing ontologies in the AEC field, either by adopting their terminology and vocabulary or by aligning with them. This section aims to analyse and compare these ontologies based on various criteria, recommend how they can enhance the AEC3PO ontology, and identify any existing gaps in these ontologies.

This section provides an overview of the existing ontologies in the AEC field. The first subsection defines our scope. The second subsection outlines the benchmarking criteria used for evaluating and comparing the ontologies. The third subsection presents a comprehensive table describing the ontologies and their respective characteristics. The fourth subsection details the DL for each ontology, providing support for reasoning tasks. The fifth subsection focuses on the metrics used to evaluate the ontologies, while the last subsection describes the evaluation methods. The concluding section highlights the advantages of this section for other tasks within the project and identifies any deficiencies or gaps in the ontologies presented.

2.1 Scope

This section contains a compilation of ontologies pertaining to the AEC domain. These ontologies have been gathered from various sources including the Semantic Building Information Modeling (BIM) library in Zotero and the building working groups and communities such as Brick Consortium[1], organisations such as BuildingSmart[2], and projects such as BIMERR[3] (BIM-based holistic tools for Energy-driven Renovation of existing Residences).

2.2 Benchmarking criteria

We define in this subsection the criteria and the metrics we used to compare and evaluate the existing ontologies. These metrics assess the usability of ontologies and indicate whether they are utilized by a broad or limited community. The ACCORD project aims at having an impact at the European scale and needs therefore to only reuse ontologies of high quality, findable, accessible, reusable, with a strong community. Knowing the publisher plays a significant role in determining the dissemination and adoption of ontologies. For instance, ontologies developed by standardization groups like International Organization for Standardization (ISO) generally have a wider audience. These metrics also consider the availability and the accessibility of each ontology. The accessibility relies on an open access evaluation, while the availability criteria mean if the ontology is published and available online in case it is open access. There are different metrics and methodologies to evaluate an ontology [7–9]. Those methodologies aim mostly to identify the strengths and weaknesses of different ontologies and identify the respected/missed best practices for the ontologies.

These metrics include:

1. **Accessibility**: according to Open Access[4], It indicates whether the work can be utilized, altered, and shared without restrictions by individuals for any purpose.
2. **Availability** online: indicates if the ontology is available online or not.
3. **FAIRness evaluation**:
   This evaluation is based on Findable, Accessible, Interoperable, Reusable (FAIR) principles[5]. We have used O'FAIRe online tool to evaluate each ontology according to FAIR four axes:
   1) **Findable**:
      The first step in (re)using data is to find them. Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, so this is an essential component of the FAIRification process.
      - F1. (Meta)data are assigned a globally unique and persistent identifier.
      - F2. Data are described with rich metadata (defined by R1 below).
      - F3. Metadata clearly and explicitly include the identifier of the data they describe.
      - F4. (Meta)data are registered or indexed in a searchable resource.
   2) **Accessible**:

[1] https://brickschema.org/consortium/
[2] https://www.buildingsmart.org/
[3] https://bimerr.eu/
[4] Open Access: http://opendefinition.org/
[5] FAIR principles: https://www.go-fair.org/fair-principles/

Once the user finds the required data, she/he/they need to know how they can be accessed, possibly including authentication and authorisation.

- A1. (Meta)data are retrievable by their identifier using a standardised communication protocol.
- A1.1 The protocol is open, free, and universally implementable.
- A1.2 The protocol allows for an authentication and authorisation procedure, where necessary.
- A2. Metadata are accessible, even when the data are no longer available.

3) **Interoperable**:
The data usually need to be integrated with other data. In addition, the data need to be interoperable with multiple applications or workflows for analysis, storage, and processing.
   - I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
   - I2. (Meta)data use vocabularies that follow FAIR principles.
   - I3. (Meta)data include qualified references to other (meta)data.
4) **Reusable**:
The goal of FAIR is to optimise the reuse of data. To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings.
   - R1. (Meta)data are richly described with a plurality of accurate and relevant attributes.
     - R1.1. (Meta)data are released with a clear and accessible data usage license.
     - R1.2. (Meta)data are associated with detailed provenance.
   - R1.3. (Meta)data meet domain-relevant community standards

4. **Evaluation with OntOlogy Pitfall Scanner![6] (OOPS!)**: this evaluation detects any bad practices during the development and the sharing of an ontology.
5. **Evaluation with OQuaRE[7] methodology**: evaluate the ontology according to OQuaRE metrics (Duque-Ramos et al., 2013) and give it a score based on:
   1) **Transferability**: it refers to the ability of an ontology to be reused or applied in different contexts or domains. An ontology should be designed in a way that allows its concepts, relationships, and rules to be easily transferred and adapted to other applications or domains.
   2) **Functional Adequacy**: it assesses whether an ontology adequately represents the intended domain or knowledge domain. It examines whether the ontology captures all the necessary concepts, relationships, and constraints required to fulfil the objectives of the ontology.
   3) **Structural Adequacy**: it evaluates the organization and coherence of an ontology. It considers the logical and hierarchical arrangement of concepts, the clarity of relationships between concepts, and the overall consistency of the ontology's structure.
   4) **Operability**: it refers to the ease of use and practicality of an ontology. It examines how well the ontology can be employed by end-users or applications, including factors such as ease of understanding, accessibility, and usability of the ontology components.
   5) **Maintainability**: it assesses how easily an ontology can be modified, extended, or updated over time. It involves factors like the clarity of ontology documentation, the

[6] OntOlogy Pitfall Scanner! web site: https://oops.linkeddata.es/index.jsp
[7] OquaRE reposetory: https://github.com/giulianodelagala/CURIOCITY/tree/master/Evaluation/OquaRE

availability of appropriate tools for ontology development and maintenance, and the ability to incorporate changes or adapt to evolving requirements.

6) **Compatibility**: it evaluates the ability of an ontology to work harmoniously with other systems, standards, or ontologies. It considers factors such as interoperability, adherence to existing standards or guidelines, and the ability to integrate with other ontologies or knowledge resources.
7) **Reliability**: it assesses the accuracy, consistency, and dependability of an ontology. It examines whether the ontology produces consistent and reliable results, conforms to logical rules and constraints, and can be trusted as a source of knowledge or information.

These characteristics are important considerations when evaluating ontologies, as they impact the ontology's usability, effectiveness, and longevity. Therefore, we will assign a rating to each metric for every ontology, allowing us to determine which ontology can be chosen as a reference for AEC3PO with a clear understanding. For each ontology, we will thus provide a rating on metric for each methodology. This will enable us to have a clear view on what ontology should be selected as reference to AEC3PO.

2.3 Analysis of ontologies related to the AEC domain

In this subsection we represent a list of existing domain ontologies utilized in the AEC field, with their characteristics. We have also classified those ontologies according to their purpose. Some ontologies are designed to model the building environment, others are proposed for automatic compliance checking, for regulations, or for building permitting. Since there are many sub-classifications for the building environment and an ontology can be designed to model more than one building characteristic, we tagged each ontology by the building characteristic it covers. We have also mentioned the publisher of each ontology, the year of development/proposition, the targeted audience, the availability and the accessibility of the ontology, the home page, and the ontology citations in scientific publications.

Several ontologies such as Building Element Ontology (BEO), Building Topology Ontology (BOT), Building Product Ontology (BPO), and Brick are implemented to model the building and its related data. Additionally, some ontologies like the Compliance Management Ontology (CoMOn), and the Post-Occupancy Evaluation Ontology (POE) are employed for automatic compliance checking of buildings. Furthermore, various regulatory ontologies including the Building Assessment Ontology (BAO), Basic Formal Ontology (BFO), the European Legislation Identifier (ELI) ontology, and the Interconnected Data Dictionary Ontology (IDDO) are developed to comply with specific regulations. Nevertheless, to the best of our knowledge, only one ontology, the Ontology for building permit authorities (OBPA), is designed for building permit applications.

Each ontology designed for building modelling focuses on one or more specific aspect of the building environment. For instance, BEO, BIMERR Material Properties ontology, the Industry Foundation Classes (IFC) Web of Data ontology (ifcWOD), and the Distribution Element Ontology (MEP) are designed to support building elements representation. Conversely, some ontologies are tailored for a single building component such as sensors, such as BIMERR Sensor Data Ontology, the Sensor, Observation, Sample, and Actuator (SOSA) ontology, and the Semantic Sensor Network (SSN) ontology. On the other hand, certain ontologies, like IFC in Web Ontology Language (ifcOWL), can represent multiple building features such as materials, geography, products, sensors, and social information including actors, persons, and organizations. Similarly, the RealEstateCore (REC) ontology models building sensors, products, ventilation and air conditioning, lighting, fire safety, and geometry. Also, Brick ontology covers multiple building characteristics, including ventilation and air conditioning, lighting, fire safety, spatial characteristics, electricity, and equipment. Although several ontologies support these features, the Quantities, Units, Dimensions and Data Types (QUDT) ontology provides precise descriptions of the measurement units.

Certain ontologies are created by the same organization or as part of the same project, like the BIMERR ontologies. The BIMERR project[8] aims to provide support to stakeholders involved in the renovation of existing buildings by developing a new toolkit. The BIMERR ontologies consist of various ontologies, such as the Annotation Objects Ontology, Building Ontology, Information Objects Ontology, the Key Performance Indicator (KPI) ontology, Material Properties Ontology, Metadata Ontology, Occupancy Profile Ontology, Renovation Process Ontology, Sensor Data Ontology, and Weather Ontology. Additionally, there are the Digital Construction Ontologies (DiCon)[9] that facilitate semantic interoperability between systems in the construction and renovation domain. DiCon comprises ten ontologies with distinct characteristics, including the agent's ontology, contexts ontology, energy ontology, entities ontology, information ontology, lifecycle ontology, materials ontology, occupancy ontology, processes ontology, and variables ontology. Some building ontologies, like the Ontology for Managing Geometry (OMG), the File Ontology for Geometry formats (FOG), BPO, the IFC Properties ontology (IFC-PROPS), Brick, and REC ontology, have been developed externally within the Linked Building Data (LBD) community[10].

The featured ontologies exhibit distinct accessibility characteristics. Some ontologies, such as BAO, the Building Circularity Assessment Ontology (BCAO), BEO, and the BIMERR ontologies, are freely accessible and available online. Conversely, other ontologies like the Domain Ontology for Construction Knowledge (DOCK), the Facility Smart Grid Information Model (FSGIM), etc., are not openly accessible. While CoMOn, the Infrastructure and Construction PROcess Ontology (IC-PRO-Onto), ifcWOD, and POE ontology have open-access papers published, they are not readily available on the web. Although BFO is not openly accessible, we managed to find it because it is aligned[11] with the DiCon ontologies.

2.3.1 Generic Built Environment Ontologies

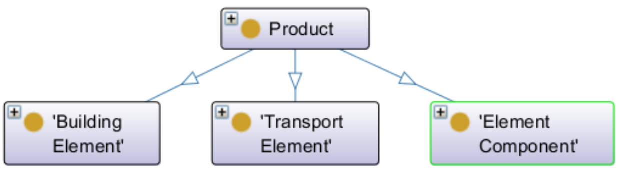


Figure 1: BEO's classification of building elements

The development of the AEC3PO ontology can draw inspiration from other ontologies that represent building information or regulatory compliance. The first group of ontologies that can serve as inspiration for the AEC3PO ontology are those designed to model the physical elements of a building (see Table 1). For example, the BEO ontology describes the physical components of a building based on the IfcBuildingElement specification and categorizes them into three groups: primary elements such as walls, beams, roofs, doors, and other structural and space-separating elements; minor items such as panels, insulation parts, glue, nails, and other items added to reinforce or connect primary elements; and transportation elements such as elevators and lifting gear. The BEO's classification

[8] BIMERR project home page: https://bimerr.eu/
[9] Digital Construction Ontologies home page: https://digitalconstruction.github.io/v/0.5/
[10] LBD external ontologies: https://github.com/w3c-lbd-cg/ontologies
[11] BFO-ISO-2018-07-24: https://buffalo.app.box.com/v/bfo-iso-owl-cl/

of building elements (shown in Figure 1) can provide an interesting reference for the AEC3PO ontology to represent the various components of a building. On the other hand, ifcOWL utilizes the complete description of IfcBuildingElement, providing a more precise classification for building elements. As depicted in Figure 2, ifcOWL offers a comprehensive description of the element representation with 12 subclassifications of elements that encompass a wide range of building-related components, from basic products to furnishing elements. The AEC3PO ontology can also draw from the building-related properties outlined in IFC4[12]. This same approach was taken by the IFC-PROPS ontology, which describes 33 different concepts representing various properties such as temporal, area, dimension, physical properties (e.g., density and thermal resistance), and more. The BOT ontology, on the other hand, focuses on building-related topology. Its most significant concept is “Zone”, which refers to a portion of the physical or virtual world with a 3D spatial extent. This zone can represent a building, a space like a room, or even a floor. The Ontology for Property Management (OPM), which is based on the Smart Energy Aware Systems (SEAS) concepts, is utilized by the OMG ontology for object geometry descriptions. This comprehensive representation can assist in the creation of the AEC3PO ontology design. The Distributed Energy Resources (DER) ontology, on the other hand, provides a less detailed geometry description based on the UML spatial schema defined in ISO 19107[13]. Lastly, the QUDT ontology provides precise descriptions of measurement units that can be used to feed AEC3PO regulation descriptions with a variety of units.

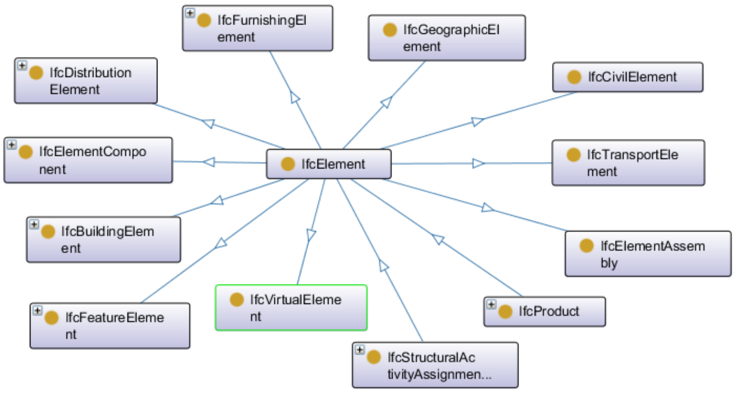


Figure 2: The element description in ifcOWL

[12] IFC standard: https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/
[13] https://www.iso.org/standard/66175.html

Table 1: Built Environment Ontologies

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
BEO	The Building Element Ontology provides an ontology based on the IfcBuildingElement subtree in the IFC specification, containing a taxonomy of classes that allow to define common building elements.	Element	Ghent University, Belgium.	2020	Yes	Yes	website	
BIMERR Annotation Objects Ontology	The Annotation Objects ontology aims to represent the annotations produced during the development of a building renovation project. These annotations serve to inform about issues or missing information that could be relevant for the project, such as indicating that a building element is missing in the BIM model.	renovation, annotation	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR[14] project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Building Ontology	This ontology aims to model building data for the BIMERR project.	topology, component	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Information Objects Ontology	This ontology aims to model the files and documents attached to building elements.	document	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)

[14] BIM-based holistic tools for Energy-driven Renovation of existing Residences: https://bimerr.eu/about/

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
BIMERR KPI Ontology	The Key Performance Indicator ontology aims to model Key Performance Indicator information related to building renovation works for the BIMERR project.	energy, renovation	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Material Properties Ontology	This ontology aims to model the properties needed to describe building elements for the BIMERR project.	element, material	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Metadata Ontology	This ontology defines annotation properties to support the ontology to data model transformation.	medata	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Occupancy Profile ontology	This ontology aims to model occupants behavior inside buildings for the BIMERR project.	occupancy	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Renovation Process Ontology	This ontology aims to model the construction processes in a building renovation project.	renovation	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Sensor Data Ontology	This ontology aims to model data from sensors located inside buildings for the BIMERR project.	sensor	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)
BIMERR Weather Ontology	This ontology aims to model weather data for the BIMERR project.	weather	Ontology Engineering Group, Universidad Politécnica de Madrid - BIMERR project	2017-2019	Yes	Yes	website	(Rasmussen et al., 2022)

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
BOT	The Building Topology Ontology is a minimal ontology for describing the core topological concepts of a building.	topology	The Vienna University of Technology and the Fraunhofer Institute for Building Physics	2013	Yes	Yes	website	(Mads Holten Rasmussen et al., 2017; Rasmussen et al., 2021, 2017a, 2017b)
BPO	The Building Product Ontology 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, as is often the case in template-driven product descriptions.	product	The University of Southern California in the United States	2008	Yes	Yes	website	
Brick	Brick contains a semantic description of the physical, logical and virtual assets in buildings and the relationships between them.	ventilation air condition, lighting, fire, spatial, electricity, equipment	The Brick Consortium, Inc.	2016	Yes	Yes	website	(Balaji et al., 2016; Gabriel Fierro et al., 2022; Jack and Wei Xi, 2019; Koh et al., 2018; Pruvost and Zeidler, 2022; Ramanathan and Husmann, 2022; Rasmussen et al., 2022)
COINS Building Information System schema	The Construction Industry Solutions (COINS) standard has a built-in structure to extend the central schema with sub-models (reference frameworks) that focus on specific topics for certain particular applications	design	Ibis, a Dutch software company	Early 2000s	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
DBO	The Digital Buildings ontology (DBO) is used by Google to represent structured information about buildings and building-installed equipment.	equipment	The National Research Council of Canada (NRC) in collaboration with industry partners, including Siemens and Autodesk	2017	Yes	Yes	website	
DCagents	Digital Construction Agents: Agents ontology formalize the the representation of the actors and stakeholders over the construction lifecycle, to support data sharing of the social, organizational and contractural relations. The ontology is aligned with BFO, Friend of a friend (FOAF) ontology and the Organization Ontology (ORG).	agent	The European Union's Seventh Framework Programme for research and innovation	2019	Yes	Yes	website	
DCAT	Data Catalog Vocabulary (DCAT) discribes RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web	product lifecycle, vocabulary	World Wide Web Consortium (W3C)	2020	Yes	Yes	website	
DCcontexts	Digital Construction Contexts: Context ontology provides the basic representation mechanisms for multi-context information in contruction and renovation projects	contruction, renovation	The European Union's Seventh Framework Programme for research and innovation	2019	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
DCenergy	Digital Construction Energy: An ontology for energy systems and energy efficiency in the construction and renovation domain	energy	The European Union's Seventh Framework Programme for research and innovation	2020	Yes	Yes	website	
DCentities	Digital Construction Entities: Digital Construction Entities Ontology defines the basic classes and properties needed for the representation of construction and renovation projects. Examples are building object, location, material batch, equipment, agent, information content entity, activity and related time concepts. The ontology is based on BFO.	construction, renovation, location, material, equipment, agent, information	The European Union's Seventh Framework Programme for research and innovation	2020	Yes	Yes	website	
DCinformation	Digital Construction Information: Digital Construction Information ontology defines the representation of information content entities in construction and renovation, including models, plans, scenarios, messages, issues, videos and point clouds. The focus is on identifiable information contents (such as first version of the architectural model of a project), not on the particular information carries (such as hard disk, cloud storage, paper print).	construction, renovation, model, plan, scenario, message, issue, video	The European Union's Seventh Framework Programme for research and innovation	2020	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
DClifecycle	Digital Construction Lifecycle: An ontology to represent the enhancement of building data throughout the construction lifecycle stages	construction	The European Union's Seventh Framework Programme for research and innovation	2020	Yes	Yes	website	
DCmaterials	Digital Construction Materials: The Material Ontology defining the main concepts of building material, type and its properties.	material	The European Union's Seventh Framework Programme for research and innovation	2021	Yes	Yes	website	
DCoccupancy	Digital Construction Occupancy: Digital Construction Occupancy ontology represents those aspects of construction and renovation projects that concern the comfort, safety and health of occupants, including visual and thermal comfort, indoor air quality and building acoustics, as well as related sensor observations.	occupancy, comfort, safety, air quality, sensor	The European Union's Seventh Framework Programme for research and innovation	2020	Yes	Yes	website	
DCprocesses	Digital Construction Processes: Process ontology for digital construction	construction	The European Union's Seventh Framework Programme for research and innovation	2019	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
DCvariables	Digital Construction Variables: The objectified property representation is orthogonal to the other definitions in the ontologies: any property can be objectified and consequently be subject of constraints.	property	The European Union's Seventh Framework Programme for research and innovation	2020	Yes	Yes	website	
DOCK	Domain Ontology for Construction Knowledge categorizes construction knowledge across three main dimensions: concept, modality, and context.	construction	the University of Florida in the United States	2007-2012	No	No		(El-Diraby, 2013)
FOG	The File Ontology for Geometry formats (FOG) provides geometry schema specific relations between things (e.g. building objects) and their geometry descriptions. These geometry descriptions can be (1) RDF-based, (2) RDF literals containing embedded geometry of existing geometry formats and (3) RDF literals containing a reference to an external geometry file.	geometry	Institute of Numerical Methods and Computer Science in Civil Engineering, University of Darmstadt, Germany. The research group Building Physics and Sustainable Design at Ghent Technology Campus. Ghent University, Belgium.	2020	Yes	Yes	website	
FSGIM	The Facility Smart Grid Information Model (FSGIM) standard is one part of a larger ecosystem of standards that support the development and implementation of a smart electric grid. The FSGIM uses Unified Modeling Language to	electricity	The National Institute of Standards and Technology (NIST) in the United States	2014	No	No		(Bushby, 2016)

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
	define key concepts that must map between electricity providers and electricity consumers							
IC-PRO-Onto	Infrastructure and Construction PROcess Ontology is proposed to offer a conceptualization and formal representation of domain process knowledge.	process	The Norwegian University of Science and Technology (NTNU) in Trondheim, Norway	2010	No	Yes		(El-Gohary et al., 2010)
ifcOWL	ifcOWL provides a Web Ontology Language (OWL) representation of the IFC schema.	material, Geographic Information System (GIS), product, sensor, social	buildingSMART International	2014	Yes	Yes	website	(Pieter Pauwels et al., 2017b; Pauwels and Roxin, 2017; Pauwels and Terkaj, 2016; Terkaj and Pauwels, 2017)
IFC-PROPS	Contains list of properties extracted from IFC4	property	Institut Mines-Télécome, France	2017	Yes	Yes	website	
ifcWOD	The IFC Web of Data ontology formally extends the ifcOWL ontology.	element	ACTIVe3D, Dijon, France. Checksem, LE2I UMR6306, CNRS, ENSAM, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France. Checksem, LE2I UMR6306, CNRS, ENSAM, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France	2015	No	Yes		(Mendes de Farias and Roxin, 2015)

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
MEP	The Distribution Element Ontology provides an ontology based on the IfcDistributionElement subtree in the IFC specification, containing a taxonomy of classes that allow to define common distribution elements (actuators, flowterminals, ...).	element	Ghent University, Belgium.	2020	Yes	Yes	website	
OMG	The Ontology for Managing Geometry (OMG) is an ontology for attaching geometry descriptions to their corresponding things (e.g. building objects). The OMG is designed to provide three levels of adding the geometry descriptions which can be used and combined flexibly.	geometry	Institute of Numerical Methods and Computer Science in Civil Engineering, University of Darmstadt, Germany. Eindhoven University of Technology, Netherlands.	2019	Yes	Yes	website	
OpenADR	The Open Advanced Demand Response (OpenADR) was created to standardize, automate, and simplify Demand Response (DR) and Distributed Energy Resources (DER) to enable utilities and aggregators to cost-effectively manage growing energy demand & decentralized energy production, and customers to control their energy future	electricity, geometry, energy	The OpenADR Alliance	2010	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
OPM	The Ontology for Property Management (OPM) is an ontology for describing temporal properties that are subject to changes as the building design evolves.	temporal properties	W3C LBD Community Group	2018	Yes	Yes	website	(Mads Holten Rasmussen et al., 2018)
QUDT	QUDT CATALOG - Quantities, Units, Dimensions and Data Types Ontologie: used to model units of measurement	unit of mesurement	QUDT.org organization, United States	2022	Yes	No	website	
REC	RealEstateCore is a modular ontology, that is, a collection of data schemas that describe concepts and relations that can occur in data that is generated to model buildings and building systems, or that is sourced from such systems.	sensor, product, ventilation air condition, lighting, fire, geometry	Agency9, a Swedish software company	2017	Yes	Yes	website	
SAREF	The Smart Applications REFerence ontology (SAREF) is intended to enable interoperability between solutions from different providers and among various activity sectors in the Internet of Things (IoT), thus contributing to the development of the global digital market. It has two extensions: SAREF4ENER for energy and SAREF4BLDG for buildings.	sensor	The European Commission's Joint Research Centre (JRC)	2015	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
SAREF4BLDG	SAREF4BLDG is the SAREF extension for building devices, and aims for a more efficient interaction and integration of actors, methods and tools during the different phases of the building life cycle.	sensor, lighting, fire	JRC	2017	Yes	Yes	website	
SEAS	The Smart Energy Aware Systems (SEAS) knowledge model is a key enabler for the semantic interoperability at the basis of SEAS use cases and business models for energy efficiency	energy	ARMINES Fayol, VTT, IMT	2017	Yes	Yes	website	(Lefrançois et al., 2017)
SOSA	The Sensor, Observation, Sample, and Actuator ontology provides a standardized vocabulary and data model for describing and sharing information about sensors, observations, samples, and actuators. It aims to address the challenges of integrating and exchanging data from diverse sources, with varying levels of granularity and complexity, by defining a set of common concepts and relationships for describing the components of an IoT system.	sensor	SSN Working Group in W3C	2017	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
SSN	The Semantic Sensor Network 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	sensor	W3C and Open Geospatial Consortium (OGC)	2011	Yes	Yes	website	

2.3.2 Building permitting ontologies

To the best of our knowledge, the only ontology within this specific category comprises OBPA (Ontology for Building Permit Authorities) (see Table 2), which pertains to building permitting. OBPA delineates the organization and framework of building permit authorities. In ACCORD context, OBPA concepts and properties can inspire AEC3PO to represent the building permit authorities and to establish links to the permits they will examine.

Table 2: Ontologies related to building permitting

Ontology	Description	Organisation	Year	Ontology available online	Open access of description	Home page
OBPA	The Ontology for building permit authorities contains vocabulary for describing organizational structures of building permit authorities.	Bauhaus-Universität Weimar, Germany	2019	Yes	Yes	website

2.3.3 Building compliance ontologies

The AEC3PO ontology can find inspiration from the ontologies outlined in Table 3, which primarily focus on the validation of building compliance. An instance of this is the BCAO ontology employed in (Morkunaite et al., 2021), which verifies building compliance with circulation regulations. The AEC3PO ontology can also utilize this ontology to establish regulations for circulation within building spaces. Moreover, the Chinese compliance ontologies offer valuable inspiration for the development of the AEC3PO ontology. For instance, the Building Regulation Ontology incorporates Chinese building regulations and utilizes information captured by IoT devices (modelled using the SSN ontology) to verify compliance with these regulations. The Code ontology describes building components, such as ramps, walls, and windows, that are subject to inspection under Chinese regulations, thus facilitating the verification of compliance. Additionally, both the Construction Quality Inspection and Evaluation (CQIE) ontology and DFS ontology are specifically designed to assess Chinese building compliance with safety regulations.

Table 3: Ontologies for building compliance

Ontology	Description	Main focus	Organisation	Year	Ontology available online	Open access of description	Home page	Citations
BAO	The Building Assessment Ontology is a concise ontology, developed to semantically describe standards, building codes, certification schemes and regulations in the AEC industry. It's aim is to make the integration of such schemes with the actual building easier, to automate the performance evaluation of buildings.	performance evaluation of buildings	NRC	2012-2015	Yes	Yes	website	
BCAO	Building Circularity Assessment Ontology is proposed to structure the scattered heterogenous manufacturer product data needed for the circulation assessment.	circulation	NRC	2018-2020	Yes	Yes		(Morkunaite et al., 2021)
Building Regulation Ontology	It is developed to represent the knowledge of building regulations. It defines generic concepts and relationships. Then, the constraints in building regulations can be modeled into OWL axioms and SPARQL rules	regulations, building environment	Huazhong University of Science and Technology, Wuhan, Hubei, China; Hubei Engineering Research Center for Virtual, Safe and Automated Construction, Wuhan, Hubei, China	2018	No	Yes		(Zhong et al., 2018)

Ontology	Description	Main focus	Organisation	Year	Ontology available online	Open access of description	Home page	Citations
Code Ontology	The code ontology designed primarily for code compliance checking is used to describe regulatory information. It is developed based on the regulatory documents for providing the semantics of the compliance checking domains.	Construction, parameters, relations, geormetric, spatial	Shanghai Jiao Tong University, Shanghai, China; Shanghai Key Laboratory for Digital Maintenance of Buildings and Infrastructure, Shanghai, China	2022	No	Yes		(Jiang et al., 2022)
CoMOn	Compliance Management Ontology is a shared conceptualization for research and practice in compliance management		The Vienna University of Technology in Austria	2010-2015	No	Yes		(Syed Abdullah et al., 2012)
CQIE Ontology	Construction Quality Inspection and Evaluation (CQIE) Ontology serves as a meta model, defining the generic terms and relations related to the construction quality compliance checking. It allows to model regulations in OWL axioms and SWRL rules for construction quality inspection and evaluation	construction, quality	Huazhong Univ. of Science and Technology, Wuhan, China; Huazhong Univ. of Science and Technology, Wuhan, China	2012	No	Yes		(Zhong et al., 2012)
DFS ontology	The Design for Safety (DFS) ontology is developed based on Natural Language Processing (NLP) for ACC Using BIM. It is designed for the chinese safety regulations.	safety, design, regulations	Nanjing Tech University, Nanjing, China; Lanzhou Jiaotong University, Lanzhou, China; Southeast University, Nanjing, China		No	Yes		(Zhou et al., 2022)
LKIF-core ontology	Legal Knowledge Interchange Format (LKIF) core ontology is designed to represent basic legal concepts for buildings.	legal concepts	ESTRELLA project	2007	Yes	Yes	website	(Hoekstra et al., 2007)

Ontology	Description	Main focus	Organisation	Year	Ontology available online	Open access of description	Home page	Citations
POE ontology	Post-Occupancy Evaluation Ontology is developed to evaluate the actual building performance against the theoretical design intents after the building has been occupied for some time.		NRC	2013-2015	No	Yes		(Zhao and Yang, 2021)
Railway Code Ontology	Semi-automatic generation of Code Ontology using ifcOWL in compliance checking	construction	Xi'an University of Technology, Xi'an, China; China Railway First Survey and Design Institute Group Co.,Ltd, Xi'an, China	2021	No	Yes		(Li et al., 2021)

2.3.1 Regulations ontologies

The AEC3PO ontology can draw inspiration from the ontologies presented in Table 4, which concentrate on building regulations that can be used in the verification of building compliance. For example, the BAO ontology provides a semantic description for building codes, standards, certification schemes, and regulations. The BCAO ontology, on the other hand, is designed to express the assessment of building circulation, which could be used by the AEC3PO ontology to create regulations for circulation within building spaces. The LKIF ontology represents the fundamental legal concepts for buildings, and the AEC3PO ontology can draw on its representation of legal resources. Finally, the ELI ontology can provide the AEC3PO ontology with a description of legislation, particularly legal resources, expressions, and formats.

The Chinese regulation and compliance ontologies can also serve as inspiration for the AEC3PO ontology. The Building Regulation Ontology, for example, is based on Chinese building regulations and the information captured by IoT devices (modelled using the SSN ontology). The Code ontology describes building elements that are subject to inspection under Chinese regulations, such as ramps, walls, and windows. The CQIEOntology and DFS ontology are both designed to support safety regulations in Chinese buildings. Additionally, the Safety Regulation Ontology (SRO) models safety regulation documents.

Table 4: Ontologies to represent regulations

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
BAO	See Table 3 for the ontology description							
BCAO	See Table 3 for the ontology description							
BFO (ISO/IEC 21838-2:2021)	Basic Formal Ontology is an ontology that is conformant to the requirements specified for top-level ontologies in ISO/IEC 21838-1. It supports the interchange of information among heterogeneous information systems.	regulations	ISO	2021	Yes	No	website	(Abdelghani, 2021; Rasmussen et al., 2022; Seppo Törmä and Yuan Zheng, 2022)
Building Regulation Ontology	See Table 3 for the ontology description							
Code Ontology	See Table 3 for the ontology description							
CQIE Ontology	See Table 3 for the ontology description							
DFS ontology	See Table 3 for the ontology description							
ELI ontology	The European Legislation Identifier ontology was designed to make legislation available online in a standardised format, so that it can be accessed, exchanged and reused across borders.	regulations	Publications Office of the European Union in collaboration with JRC	2010	Yes	Yes	website	

Ontology	Description	Main focus	Organisation	Year	Available online	Open access	Home page	Citations
IDDO	The Interconnected Data Dictionary Ontology maps the data model of the ISO 23386[15] and establishes additional classes for linking properties.	regulations	Ruhr University Bochum	Last updated in 2015	Yes	Yes	website	(Zentgraf et al., 2022)
LKIF-core ontology	See Table 3 for the ontology description							
SRO	The objective of the Safety Regulation Ontology (SRO) is to extract the fundamental ideas and semantic connections from safety restrictions in construction, which are presented in various regulations, at the level of individual sentences. These safety constraints can be expressed as OWL axioms and SPARQL.	safty regulations, riscks,	School of Civil Engineering, Southeast University, Nanjing, China; Eindhoven University of Technology, Eindhoven, the Netherlands	2022	No	Yes		(Li et al., 2022)

[15] https://www.iso.org/standard/75401.html

2.4 A Comparison of Ontologies Using DL

In this subsection, we compare the ontologies based on their supported DL and OWL profile (see Table 5). Initially, we employed the DLExpressivityChecker[16] method from the OWLAPI Java library to determine the basic logics and extensions of DL supported by each ontology. This method enumerated all the logic prefixes that describe DL (AL, C, D, E, EL, ELPLUSPLUS, F, H, I, N, O, Q, R, S, TRAN, U). Next, we defined the DL based on the constructor restrictions, including individual expressions, property expressions, and class expressions. Finally, we compared each ontology to the default OWL profiles and tested for compatibility with them. We verified the compatibility with the five OWL 2 profiles: OWL 2 DL, OWL 2 EL, OWL 2 QL, and OWL 2 RL. In cases where the ontology profile is found to be consistent with one of the default OWL 2 profiles, the true value is attributed, while in profiles where it is not compatible with, the false value is attributed instead. Furthermore, all the listed ontologies are compatible with the OWL 2 FULL profile.

Analysing the DL aspect of ontologies is important for several reasons. DL provides a formal and rigorous way to define the concepts and relationships within an ontology. This helps to ensure consistency and completeness of the ontology, as well as providing a clear and precise understanding of the meaning of the concepts and relationships. By comparing the DL of different ontologies, we can identify similarities and differences in their conceptualizations of the domain and determine which ontologies might be suitable for interoperability and integration with each other. Additionally, DL can be used to check the ability to support reasoning for the ontologies, answering queries and inferring new knowledge based on the existing concepts and relationships. This enables the automated processing of the ontology, which is useful for the ACC task.

[16] DLExpressivityChecker home page: http://owlcs.github.io/owlapi/apidocs_3/org/semanticweb/owlapi/util/DLExpressivityChecker.Construct.html

Table 5: DL for the ontologies

Ontology	DL basic logics and extensions	DL Based Constructs	OWL 2 DL	OWL 2 EL	OWL 2 QL	OWL 2 RL
BAO	RRESTRCRIF(D)	SROIQ	false	false	false	false
BCAO	RRESTRCUUNIVRESTREHN(D)	SROIQ	true	false	false	false
BEO	(D)	ALC	true	false	false	true
BIMERRAO	RRESTRUNIVRESTRN(D)	EL	true	false	false	false
BIMERRB	RRESTRCUNIVRESTREQ(D)	EL	true	false	false	false
BIMERRIO	RRESTRN(D)	EL	true	false	false	false
BIMERRKPI	RRESTRUNIVRESTRQ(D)	EL	true	false	false	false
BIMERRM		OWL2 Full	true	true	true	true
BIMERRMP	RRESTRUNIVRESTRQ(D)	EL	true	false	false	false
BIMERROP	RRESTRCUUNIVRESTREON(D)	ALC	true	false	false	false
BIMERRRP	RRESTRUNIVRESTRN(D)	EL	true	false	false	false
BIMERRSD	RRESTRUNIVRESTRQ(D)	EL	true	false	false	false
BIMERRW	RRESTREHN(D)	ALC	true	false	false	false
BOT	RRESTRCRI	ALC	true	false	false	true
BPO	RRESTRCCINTRIN(D)	ALC	false	false	false	false
Brick		ALC	true	true	true	true

Ontology	DL basic logics and extensions	DL Based Constructs	OWL 2 DL	OWL 2 EL	OWL 2 QL	OWL 2 RL
COINS	RRESTRCUCINTUNIVRESTRHIQ(D)	ALC	false	false	false	false
DBO	CUCINTUNIVRESTREH+(D)	ALC	false	false	false	false
DCagents	RRESTRCCINTUNIVRESTREHI(D)	SROIQ	false	false	false	false
DCAT	RRESTRUUNIVRESTRRrN(D)	ALC	false	false	false	false
DCcontexts	RRESTRHI	SHIQ	false	false	false	false
DCenergy	RRESTRUEHOI(D)	ALC	false	false	false	false
DCentities	RRESTRCUCINTUNIVRESTREH+IF(D)	SROIQ	false	false	false	false
DCinformation	RRESTRUNIVRESTRHI(D)	ALC	false	false	false	false
DClifecycle	RRESTRRI	SROIQ	false	false	false	false
DCmaterials	RRESTRCEROI(D)	ALC	false	false	false	false
DCoccupancy	RRESTRCINTUNIVRESTREHO(D)	SROIQ	false	false	false	false
DCprocesses	RRESTRCINTUNIVRESTREHOI(D)	SROIQ	false	false	false	false
DCvariables	RRESTRCUUNIVRESTRHI(D)	ALC	false	false	false	false
ELI ontology	RRESTRUCINTEHOIN(D)	ALC	false	false	false	false
FOG	RRESTRHF(D)	ALC	true	false	false	true
IDDO	RRESTRHQ(D)	ALC	false	false	false	false
ifcOWL	RRESTRCUUNIVRESTREIQ	ALC	false	false	false	false
IFCPROP	RRESTRUNIVRESTR(D)	OWL2 Full	false	false	false	false

Ontology	DL basic logics and extensions	DL Based Constructs	OWL 2 DL	OWL 2 EL	OWL 2 QL	OWL 2 RL
LKIF	H+I	SROIQ	true	false	false	true
MEP	O	ALC	true	false	false	true
OBPA	RRESTRCUUNIVRESTREHIF(D)	ALC	true	false	false	false
OMG	RRESTRCRIF(D)	ALC	false	false	false	false
OpenADR	RRESTRCUNIVRESTREHI(D)	ALC	false	false	false	false
OPM	RRESTRCH(D)	ALC	true	false	false	true
REC	RRESTRUNIVRESTREHIF(D)	SROIQ	false	false	false	false
SAREF	RRESTRCUUNIVRESTREIQ(D)	ALC	true	false	false	false
SAREF4BLDG	RRESTRCUUNIVRESTR+IQ(D)	EL	true	false	false	false
SEAS	RRESTRCRIF(D)	ALC	true	false	false	true
SOSA	RRESTRI(D)	ALC	true	false	false	true
SSN	UNIVRESTRRIN(D)	ALC	true	false	false	false

2.5 Ontologies metrics

This subsection is dedicated to the metrics used for evaluating ontologies. To assess the richness of the ontologies, we relied on the metrics defined by BioPortal[17]. These metrics encompass statistical measures as well as quality-control and quality-assurance metrics. The statistical metrics describe the number of classes, individuals, and properties, as well as the maximum depth, maximum number of children, and average number of children. In contrast, the quality-control and quality-assurance metrics focus on the number of classes with only one subclass, classes with over 25 subclasses, and classes with no definition.

The statistical metrics used by BioPortal to evaluate ontologies are similar to those used in Protégé[18], such as class count and individual count, which measure the richness of ontologies in terms of the number of classes and individuals. Additionally, BioPortal incorporates quality metrics, such as classes without definitions, to gain insight into the ontology's quality. These metrics can be utilized to compare and rank different ontologies, which can aid in selecting the most suitable ontology to align with the AEC3PO ontology.

We notice that some ontologies are very rich with concepts and individuals such as Brick (with 1452 class and 2566 individual), ifcOWL (with 1326 class and 1162). However, other ontologies contain less individual and less classes like ELI ontology (with 33 class and 28 individual) and BAO (with 15 class and 2 individual). The reason behind this difference is the nature of the data represented by the ontologies, for example, the ontologies that represent the building model need to contain more classes to express all or most of the building components (which is the case with ifcOWL). If the ontology represents more than one feature like the sensors, the geometric information, etc. it will need more classes to express them. For example, QUDT ontology only contains 31 class because it was designed only for units and dimensions.

[17] BioPortal metrics home page: Ontology_Metrics
[18] http://protegeproject.github.io/protege/views/ontology-metrics/

Ontology	Statistical Metrics						Quality-Control and Quality-Assurance Metrics		
	Number of classes	Number of individuals	Number of properties	Maximum depth	Maximum Number Of Siblings	Average Number Of Siblings	Classes with only one subclass	Classes with more than 25 subclasses	Classes with no definition
BAO	15	2	36	1	7	3	1	0	0
BCAO	37	0	36	1	10	6	0	0	37
BEO	**186**	1	1	4	20	5	1	0	4
BFO (ISO/IEC 21838-2:2021)	**143**	8	**193**	8	14	3	14	0	49
BIMERR Annotation Objects Ontology	7	0	30	0	6	6	0	0	1
BIMERR Building Ontology	46	0	34	6	11	2	8	0	1
BIMERR Information Objects Ontology	1	0	17	0	1	1	1	0	0
BIMERR KPI Ontology	19	0	39	1	15	5	2	0	1
BIMERR Material Properties Ontology	**140**	0	21	3	26	6	0	1	**128**
BIMERR Metadata Ontology	**140**	0	21	3	26	6	0	1	**128**
BIMERR Occupancy Profile ontology	**67**	46	**63**	3	25	3	5	1	0
BIMERR Renovation Process Ontology	12	0	43	0	11	11	0	0	4

Ontology	Statistical Metrics						Quality-Control and Quality-Assurance Metrics		
	Number of classes	Number of individuals	Number of properties	Maximum depth	Maximum Number Of Siblings	Average Number Of Siblings	Classes with only one subclass	Classes with more than 25 subclasses	Classes with no definition
BIMERR Sensor Data Ontology	10	0	16	2	6	2	3	0	0
BIMERR Weather Ontology	31	**79**	27	4	9	2	5	0	2
BOT	10	5	17	1	6	5	0	0	3
BPO	25	0	28	2	12	2	6	0	11
Brick	**1452**	**2566**	**107**	7	**102**	3	**115**	6	**476**
COINS for BIM	43	6	44	6	11	2	6	0	3
DBO	**1334**	35	7	6	**401**	14	**64**	29	**436**
DCagents	**159**	11	**269**	8	14	2	15	0	39
DCAT	**52**	18	**106**	2	14	3	3	0	11
DCcontexts	6	0	19	0	6	6	0	0	1
DCenergy	**325**	**53**	**380**	8	18	3	26	0	**112**
DCentities	**107**	8	**193**	8	14	2	13	0	13
DCinformation	**228**	15	**337**	8	15	2	24	0	**59**
DClifecycle	**167**	11	**290**	8	14	2	15	0	44
DCmaterials	**178**	**97**	**269**	8	14	3	15	0	48

Ontology	Statistical Metrics						Quality-Control and Quality-Assurance Metrics		
	Number of classes	Number of individuals	Number of properties	Maximum depth	Maximum Number Of Siblings	Average Number Of Siblings	Classes with only one subclass	Classes with more than 25 subclasses	Classes with no definition
DCoccupancy	**302**	18	**352**	9	15	2	36	0	**129**
DCprocesses	**130**	11	**233**	8	14	2	16	0	25
DCvariables	27	8	30	3	10	2	3	0	6
ELI ontology	33	28	**109**	3	10	2	11	0	9
FOG	3	4	**133**	0	3	3	0	0	2
IDDO	44	7	**72**	1	39	2	19	1	28
ifcOWL	**1326**	**1162**	**1596**	9	**208**	4	150	8	**1326**
IFC-PROPS	34	0	**596**	0	34	34	0	1	34
LKIF	**154**	0	**96**	7	8	2	29	0	1
MEP	**484**	2	0	4	23	6	1	0	**199**
OBPA	49	13	40	3	19	3	5	0	34
OMG	8	0	19	1	6	3	1	0	4
OpenADR	**77**	6	**98**	3	24	3	5	0	6
OPM	17	1	12	2	14	5	1	0	8
QUDT	31	8	31	3	26	3	6	1	6
REC	**882**	0	**518**	8	**72**	3	**51**	4	**793**

Ontology	Statistical Metrics						Quality-Control and Quality-Assurance Metrics		
	Number of classes	Number of individuals	Number of properties	Maximum depth	Maximum Number Of Siblings	Average Number Of Siblings	Classes with only one subclass	Classes with more than 25 subclasses	Classes with no definition
SAREF	**81**	10	40	3	13	4	2	0	0
SAREF4BLDG	**71**	0	**262**	6	18	4	3	0	0
SEAS	**586**	**68**	**376**	10	**58**	4	40	4	22
SOSA	16	1	23	0	16	16	0	0	3
SSN	23	2	38	1	16	4	2	0	4

2.6 Ontology evaluation tools

In this subsection we assessed the quality of each ontology using a range of evaluation tools that employ diverse metrics. The first tool, called O'FAIRe, uses FAIR Principles to evaluate the ontologies based on their findability, accessibility, interoperability, and reusability. The second tool, OntOlogy Pitfall Scanner! (OOPS!), compares the ontologies with best practices for developing, representing, and sharing ontologies on the web. Finally, we utilized the OQuaRE methodology to evaluate the ontologies based on their operability, reliability, compatibility, transferability, functional adequacy, structural soundness, and maintainability.

The evaluation results from O'FAIRe assigned a score to each of the four FAIR axes described in the Benchmarking criteria section. These results are graphicly depicted and can be accessed online for each ontology that was evaluated.

The OOPS! evaluation identified the presence of bad practices and warnings in the evaluated ontologies. It also provided information on the frequency of occurrence (the number of cases) and severity of each warning case.

The evaluation results from OQuaRE methodology has also gave a score to each characteristic including transferability, functional adequacy, structural, operability, maintainability, compatibility, and reliability. These characteristics are described in Benchmarking criteria as follow:

1. The **transferability** refers to an ontology's capacity for reuse or application in various domains or contexts. A low transferability score implies that the ontology is limited to its original context and cannot be utilized elsewhere.
2. The **functional adequacy** score measures the ontology's comprehensiveness in terms of concepts, relationships, and constraints necessary to fulfil its objectives and accurately represent the domain.
3. The **structural** characteristic assesses the clarity of relationships between concepts and the overall coherence and consistency of the ontology's structure.
4. The **operability** evaluates the accessibility of an ontology, with a higher score indicating greater usability of its components.
5. The **maintainability** reflects how easily an ontology can be modified, extended, or updated over time, considering factors like the clarity of documentation and the availability of suitable tools for development and maintenance.
6. The **compatibility** evaluates an ontology's ability to integrate with other ontologies or knowledge resources, taking into account factors such as interoperability and adherence to existing standards or guidelines.
7. The **reliability** score is determined by the accuracy, consistency, and dependability of an ontology.

BAO:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity

Results for P11: Missing domain or range in properties.	8 cases	Important
Results for P30: Equivalent classes not explicitly declared.	1 case	Important

- Evaluation with OQuaRE methodology: Result

BCAO:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P37: Ontology not available on the Web.	Ontology	Critical
Results for P37: Ontology not available on the Web.	Ontology	Critical

- Evaluation with OQuaRE methodology: Result

BEO:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	4 cases	Minor
Results for P08: Missing annotations.	5 cases	Minor
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	1 case	Important
Results for P20: Misusing ontology annotations.	59 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P30: Equivalent classes not explicitly declared.	4 cases	Important
Results for P32: Several classes with the same label.	11 cases	Minor

- Evaluation with OQuaRE methodology: Result

BFO (ISO/IEC 21838-2:2021):

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor

- Evaluation with OQuaRE methodology: NONE

BIMERR Annotation Objects Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	10 cases	Important
Results for P13: Inverse relationships not explicitly declared.	9 cases	Minor

- Evaluation with OQuaRE methodology: NONE

BIMERR Building Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P11: Missing domain or range in properties.	11 cases	Important
Results for P13: Inverse relationships not explicitly declared.	15 cases	Minor
Results for P24: Using recursive definitions.	1 case	Important
SUGGESTION: symmetric or transitive object properties.	2 cases	

- Evaluation with OQuaRE methodology: Result

BIMERR Information Objects Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P11: Missing domain or range in properties.	17 cases	Important
Results for P13: Inverse relationships not explicitly declared.	7 cases	Minor

- Evaluation with OQuaRE methodology: NONE

BIMERR KPI Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	17 cases	Important
Results for P12: Equivalent properties not explicitly declared.	1 case	Important
Results for P13: Inverse relationships not explicitly declared.	17 cases	Minor
Results for P24: Using recursive definitions.	1 case	Important
SUGGESTION: symmetric or transitive object properties.	1 case	

- Evaluation with OQuaRE methodology: Result

BIMERR Material Properties Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P07: Merging different concepts in the same class.	16 cases	Minor
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	21 cases	Important
Results for P13: Inverse relationships not explicitly declared.	13 cases	Minor

- Evaluation with OQuaRE methodology: Result

BIMERR Metadata Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
This ontology does not contain any bad practice detectable by OOPS!		

- Evaluation with OQuaRE methodology: NONE

BIMERR Occupancy Profile ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	2 cases	Minor
Results for P11: Missing domain or range in properties.	22 cases	Important
Results for P13: Inverse relationships not explicitly declared.	32 cases	Minor
Results for P24: Using recursive definitions.	1 case	Important
SUGGESTION: symmetric or transitive object properties.	2 cases	

- Evaluation with OQuaRE methodology: Result

BIMERR Renovation Process Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	38 cases	Important
Results for P13: Inverse relationships not explicitly declared.	12 cases	Minor
Results for P30: Equivalent classes not explicitly declared.	1 case	Important

- Evaluation with OQuaRE methodology: NONE

BIMERR Sensor Data Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	1 case	Minor
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	7 cases	Important
Results for P13: Inverse relationships not explicitly declared.	6 cases	Minor

- Evaluation with OQuaRE methodology: Result

BIMERR Weather Ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P08: Missing annotations.	1 case	Minor

Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	25 cases	Important
Results for P13: Inverse relationships not explicitly declared.	18 cases	Minor

- Evaluation with OQuaRE methodology: Result

BOT:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	1 case	Minor
Results for P08: Missing annotations.	1 case	Minor
Results for P11: Missing domain or range in properties.	7 cases	Important
Results for P13: Inverse relationships not explicitly declared.	12 cases	Minor
Results for P20: Misusing ontology annotations.	1 case	Minor
Results for P34: Untyped class.	2 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	1 case	

- Evaluation with OQuaRE methodology: Result

BPO:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	5 cases	Minor
Results for P07: Merging different concepts in the same class.	1 case	Minor
Results for P08: Missing annotations.	14 cases	Minor

Results for P11: Missing domain or range in properties.	9 cases	Important
Results for P13: Inverse relationships not explicitly declared.	13 cases	Minor
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	2 cases	Important
SUGGESTION: symmetric or transitive object properties.	2 cases	

- Evaluation with OQuaRE methodology: Result

Brick:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	115 cases	Minor
Results for P04: Creating unconnected ontology elements.	8 cases	Minor
Results for P08: Missing annotations.	126 cases	Minor
Results for P11: Missing domain or range in properties.	48 cases	Important
Results for P13: Inverse relationships not explicitly declared.	17 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P30: Equivalent classes not explicitly declared.	3 cases	Important
Results for P31: Defining wrong equivalent classes.	6 cases	Critical
Results for P34: Untyped class.	12 cases	Important
Results for P35: Untyped property.	54 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
Results for P40: Namespace hijacking.	4 cases	Critical
SUGGESTION: symmetric or transitive object properties.	2 cases	

- Evaluation with OQuaRE methodology: Result

COINS Building Information System schema:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P11: Missing domain or range in properties.	2 cases	Important
Results for P13: Inverse relationships not explicitly declared.	19 cases	Minor
Results for P20: Misusing ontology annotations.	17 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P24: Using recursive definitions.	31 cases	Important
Results for P38: No OWL ontology declaration.	Ontology	Important
SUGGESTION: symmetric or transitive object properties.	1 case	

- Evaluation with OQuaRE methodology: Result

DBO:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	9 cases	Minor
Results for P08: Missing annotations.	412 cases	Minor
Results for P11: Missing domain or range in properties.	7 cases	Important
Results for P13: Inverse relationships not explicitly declared.	6 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P30: Equivalent classes not explicitly declared.	11 cases	Important
Results for P40: Namespace hijacking.	1 case	Critical

- Evaluation with OQuaRE methodology: Result

DCagents:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	1 case	Minor
Results for P05: Defining wrong inverse relationships.	1 case	Critical
Results for P08: Missing annotations.	11 cases	Minor
Results for P11: Missing domain or range in properties.	26 cases	Important
Results for P12: Equivalent properties not explicitly declared.	3 cases	Important
Results for P13: Inverse relationships not explicitly declared.	127 cases	Minor
Results for P24: Using recursive definitions.	17 cases	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	38 cases	

- Evaluation with OQuaRE methodology: Result

DCAT:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 198.0 (41.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor

- Evaluation with OQuaRE methodology: Result

DCcontexts:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 187.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	3 cases	Important
Results for P13: Inverse relationships not explicitly declared.	7 cases	Minor
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	6 cases	

- Evaluation with OQuaRE methodology: NONE

DCenergy:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 187.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	1 case	Minor
Results for P04: Creating unconnected ontology elements.	1 case	Minor
Results for P05: Defining wrong inverse relationships.	3 cases	Critical
Results for P08: Missing annotations.	22 cases	Minor
Results	Number of cases	Severity
Results for P11: Missing domain or range in properties.	65 cases	Important
Results for P12: Equivalent properties not explicitly declared.	4 cases	Important
Results for P13: Inverse relationships not explicitly declared.	172 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P24: Using recursive definitions.	18 cases	Important
Results for P25: Defining a relationship as inverse to itself.	1 case	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical

Results for P30: Equivalent classes not explicitly declared.	7 cases	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	43 cases	

- Evaluation with OQuaRE methodology: Result

DCentities:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P05: Defining wrong inverse relationships.	1 case	Critical
Results for P08: Missing annotations.	5 cases	Minor
Results for P11: Missing domain or range in properties.	17 cases	Important
Results for P12: Equivalent properties not explicitly declared.	2 cases	Important
Results for P13: Inverse relationships not explicitly declared.	112 cases	Minor
Results for P24: Using recursive definitions.	17 cases	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	29 cases	

- Evaluation with OQuaRE methodology: Result

DCinformation:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 187.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	1 case	Minor
Results for P05: Defining wrong inverse relationships.	3 cases	Critical
Results for P08: Missing annotations.	17 cases	Minor
Results for P11: Missing domain or range in properties.	33 cases	Important
Results for P12: Equivalent properties not explicitly declared.	4 cases	Important
Results for P13: Inverse relationships not explicitly declared.	157 cases	Minor
Results for P24: Using recursive definitions.	18 cases	Important
Results for P25: Defining a relationship as inverse to itself.	1 case	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	4 cases	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	43 cases	

- Evaluation with OQuaRE methodology: [Result]

DClifecycle:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 189.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	1 case	Minor
Results for P05: Defining wrong inverse relationships.	1 case	Critical
Results for P08: Missing annotations.	13 cases	Minor

Results for P11: Missing domain or range in properties.	33 cases	Important
Results for P12: Equivalent properties not explicitly declared.	3 cases	Important
Results for P13: Inverse relationships not explicitly declared.	132 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P24: Using recursive definitions.	17 cases	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	38 cases	

- Evaluation with OQuaRE methodology: [Result]

DCmaterials:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 189.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P05: Defining wrong inverse relationships.	1 case	Critical
Results for P08: Missing annotations.	5 cases	Minor
Results for P11: Missing domain or range in properties.	82 cases	Important
Results for P12: Equivalent properties not explicitly declared.	2 cases	Important
Results for P13: Inverse relationships not explicitly declared.	121 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P24: Using recursive definitions.	17 cases	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical

Results for P35: Untyped property.	7 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	29 cases	

- Evaluation with OQuaRE methodology: [Result]

DCoccupancy:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 187.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	1 case	Minor
Results for P05: Defining wrong inverse relationships.	3 cases	Critical
Results for P08: Missing annotations.	18 cases	Minor
Results for P11: Missing domain or range in properties.	35 cases	Important
Results for P12: Equivalent properties not explicitly declared.	4 cases	Important
Results for P13: Inverse relationships not explicitly declared.	169 cases	Minor
Results for P24: Using recursive definitions.	18 cases	Important
Results for P25: Defining a relationship as inverse to itself.	1 case	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	4 cases	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	43 cases	

- Evaluation with OQuaRE methodology: [Result]

DCprocesses:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P05: Defining wrong inverse relationships.	1 case	Critical
Results for P08: Missing annotations.	5 cases	Minor
Results for P11: Missing domain or range in properties.	21 cases	Important
Results for P12: Equivalent properties not explicitly declared.	2 cases	Important
Results for P13: Inverse relationships not explicitly declared.	120 cases	Minor
Results for P24: Using recursive definitions.	17 cases	Important
Results for P29: Defining wrong transitive relationships.	2 cases	Critical
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P31: Defining wrong equivalent classes.	1 case	Critical
Results for P35: Untyped property.	3 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	38 cases	

- Evaluation with OQuaRE methodology: Result

DCvariables:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P11: Missing domain or range in properties.	5 cases	Important
Results for P13: Inverse relationships not explicitly declared.	14 cases	Minor
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	1 case	

- Evaluation with OQuaRE methodology: Result

ELI ontology:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	6 cases	Minor
Results for P08: Missing annotations.	13 cases	Minor
Results for P11: Missing domain or range in properties.	23 cases	Important
Results for P13: Inverse relationships not explicitly declared.	34 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	24 cases	

- Evaluation with OQuaRE methodology: Result

FOG:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P11: Missing domain or range in properties.	130 cases	Important
Results for P13: Inverse relationships not explicitly declared.	13 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P34: Untyped class.	2 cases	Important
Results for P35: Untyped property.	5 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor

Results for P40: Namespace hijacking.	2 cases	Critical

- Evaluation with OQuaRE methodology: NONE

IDDO:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 189.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P08: Missing annotations.	5 cases	Minor
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	3 cases	Important
Results for P13: Inverse relationships not explicitly declared.	29 cases	Minor
Results for P19: Defining multiple domains or ranges in properties.	19 cases	Critical
Results for P20: Misusing ontology annotations.	2 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P34: Untyped class.	11 cases	Important
Results for P35: Untyped property.	2 cases	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	7 cases	

- Evaluation with OQuaRE methodology: NONE

ifcOWL:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 187.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!): NONE
- Evaluation with OQuaRE methodology: [Result]

IFC-PROPS:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 187.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor

- Evaluation with OQuaRE methodology: NONE

LKIF:

- Evaluation with O'FAIRe (FAIRness evaluation): [Total score : 189.0 (39.0%)]
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P02: Creating synonyms as classes.	4 cases	Minor
Results for P04: Creating unconnected ontology elements.	8 cases	Minor
Results for P07: Merging different concepts in the same class.	1 case	Minor
Results for P08: Missing annotations.	247 cases	Minor
Results for P11: Missing domain or range in properties.	96 cases	Important
Results for P12: Equivalent properties not explicitly declared.	1 case	Important
Results for P13: Inverse relationships not explicitly declared.	14 cases	Minor
Results for P24: Using recursive definitions.	9 cases	Important
Results for P25: Defining a relationship as inverse to itself.	9 cases	Important
Results for P26: Defining inverse relationships for a symmetric one.	9 cases	Important
Results for P30: Equivalent classes not explicitly declared.	9 cases	Important
Results for P31: Defining wrong equivalent classes.	2 cases	Critical
Results for P41: No license declared.	Ontology	Important

Evaluation with OQuaRE methodology: [Result]

MEP:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor

- Evaluation with OQuaRE methodology: Result

OBPA:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor
Results for P37: Ontology not available on the Web.	Ontology	Critical

- Evaluation with OQuaRE methodology: Result

OMG:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	1 case	Minor
Results for P08: Missing annotations.	6 cases	Minor
Results for P11: Missing domain or range in properties.	10 cases	Important
Results for P13: Inverse relationships not explicitly declared.	19 cases	Minor
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	5 cases	

- Evaluation with OQuaRE methodology: Result

OpenADR:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	2 cases	Minor
Results for P08: Missing annotations.	1 case	Minor
Results for P11: Missing domain or range in properties.	60 cases	Important
Results for P13: Inverse relationships not explicitly declared.	55 cases	Minor
Results for P22: Using different naming conventions in the ontology.	Ontology	Minor
Results for P24: Using recursive definitions.	1 case	Important
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P36: URI contains file extension.	Ontology	Minor
SUGGESTION: symmetric or transitive object properties.	1 case	
Results for P04: Creating unconnected ontology elements.	2 cases	Minor

- Evaluation with OQuaRE methodology: Result

OPM:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	10 cases	Minor
Results for P08: Missing annotations.	10 cases	Minor
Results for P11: Missing domain or range in properties.	8 cases	Important
Results for P13: Inverse relationships not explicitly declared.	8 cases	Minor

- Evaluation with OQuaRE methodology: Result

QUDT:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor

- Evaluation with OQuaRE methodology: NONE

REC:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 218.0 (45.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor
Results for P37: Ontology not available on the Web.	Ontology	Critical

- Evaluation with OQuaRE methodology: Result

SAREF:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 192.0 (40.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P37: Ontology not available on the Web.	Ontology	Critical
Results for P38: No OWL ontology declaration.	Ontology	Important

- Evaluation with OQuaRE methodology: Result

SAREF4BLDG:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 194.0 (40.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	2 cases	Minor
Results for P11: Missing domain or range in properties.	262 cases	Important
Results for P13: Inverse relationships not explicitly declared.	175 cases	Minor
Results for P24: Using recursive definitions.	2 cases	Important
Results for P32: Several classes with the same label.	2 cases	Minor

- Evaluation with OQuaRE methodology: Result

SEAS:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 189.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P36: URI contains file extension.	Ontology	Minor

- Evaluation with OQuaRE methodology: NONE

SOSA:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 187.0 (39.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	16 cases	Minor
Results for P08: Missing annotations.	3 cases	Minor
Results for P10: Missing disjointness.	Ontology	Important

Results for P11: Missing domain or range in properties.	23 cases	Important
Results for P13: Inverse relationships not explicitly declared.	3 cases	Minor
Results for P30: Equivalent classes not explicitly declared.	1 case	Important
Results for P38: No OWL ontology declaration.	Ontology	Important
Results for P04: Creating unconnected ontology elements.	16 cases	Minor

- Evaluation with OQuaRE methodology: NONE

SSN:

- Evaluation with O'FAIRe (FAIRness evaluation): Total score : 256.0 (53.0%)
- Evaluation with OOPS! (OntOlogy Pitfall Scanner!):

Results	Number of cases	Severity
Results for P04: Creating unconnected ontology elements.	3 cases	Minor
Results for P08: Missing annotations.	3 cases	Minor
Results for P10: Missing disjointness.	Ontology	Important
Results for P11: Missing domain or range in properties.	38 cases	Important
Results for P13: Inverse relationships not explicitly declared.	3 cases	Minor
Results for P24: Using recursive definitions.	1 case	Important
Results for P30: Equivalent classes not explicitly declared.	2 cases	Important
Results for P38: No OWL ontology declaration.	Ontology	Important

- Evaluation with OQuaRE methodology: Result

2.7 Summary

We focus in this section on ontologies that pertain to buildings. These ontologies serve various purposes, including modelling buildings themselves, conducting building compliance checks, and representing building regulations. The main objective of this section is to get a clear idea about the existing ontologies in AEC field and how they can be used in ACCORD context, i.e., what modules within these ontologies can serve Task 2.2 (Compliance Ontology) during the design of AEC3PO ontology. Also, we identified the gaps in these ontologies and how AEC3PO can cover them and attend the goal of the project.

To the best of our knowledge the only ontology related to the building permitting is OBPA, however it only describes organizational structure of building permit authorities with no further details for the building permit or its characteristics. Also, according to our study contacted toward the existing ontologies, there is no ontology that link the physical description of the building (that can include, its special characteristics, design, elements, etc) and the regulations (that can include description of legislation, legal resources, safety regulation, etc.), i.e. no ontology can be used by itself to conclude on the building compliance according to its semantic description.

As part of this deliverable, we have thoroughly examined, categorized, and assessed the listed ontologies. The findings from this study can provide valuable input for the T2.2 task (Compliance Ontology) and aid in the selection of ontologies relevant to building compliance checks or general building modelling. This information can inspire the design process of AEC3PO by either reusing certain modules from the identified ontologies or aligning them accordingly.

Upon conducting evaluations using O'FAIRe, OOPS!, and OQuaRE methodologies for each ontology, the results indicate their commendable performance, making them suitable candidates for implementation in the ACCORD context to inspire AEC3PO.

3. Query Languages

Query languages play an essential role in BIM world, where data is at the heart of the project. To effectively manage and use BIM data, specialized query languages have been developed to provide a structured way to extract and manipulate information from BIM models.

BIM projects require a high level of collaboration and coordination among project stakeholders, which is facilitated using specialized query languages. In this context, query languages are essential tools for enabling efficient and effective communication and data exchange between project participants, and for unlocking the full potential of BIM technology in the construction industry.

Query languages can fulfil two main functions within the AEC domain, particularly in the ACCORD context. Firstly, they can be utilized to retrieve data. Secondly, they can be employed to express regulations and identify data that may contravene these regulations. Authors in (Bouzidi et al., 2013, 2012; Yurchyshyna et al., 2010, 2008) have formalised the rule information directly into SPARQL queries (SPARQL Protocol and RDF Query Language).

In this section, we will present a list of query languages that enable the interrogation of building data described in an RDF format or in a BIM format, alongside with geographic data.

3.1 Scope

This section mainly targets query languages associated with the AEC domain and the semantic web. These languages have been gathered from World Wide Web Consortium (W3C), Open Geospatial Consortium (OGC), and the Zotero library of the project.

3.2 List of query languages

The following list includes the main query languages that can be used for querying and processing BIM, including geospatial data:

- BIM Query Language (BIMQL) (Nawari, 2018): based on Structured Query Language (SQL).

- BimSPARQL (Hu et al., 2021) (SPARQL with BIM-specific functions)
- GeoSPARQL (buildingSmart Regulatory Room, 2017; Hu et al., 2021)
- Query Language for BIM (QL4BIM) (Solihin et al., 2018)
- Regulatory Knowledge Query Language (RKQL) (Dimyadi et al., 2016)
- SPARQL for RDF format (buildingSmart Regulatory Room, 2017; Dimyadi et al., 2016; P. Pauwels et al., 2017; Pauwels and Zhang, 2015)
- SPARQL over binary engineering data (Krijnen and Beetz, 2018; Krijnen, 2019) (e.g. HDF5 representation of IFC)

3.3 Synthesis

Query languages are essential in the interrogation and manipulation of data, as there are various data types and format. To cover different types of data without losing information, multiple query languages are proposed and developed. For the semantic data represented as RDF triplets that relies on semantic relations among data, SPARQL (buildingSmart Regulatory Room, 2017; Hoffmann et al., 2021; Krijnen and Beetz, 2018; Zhang and Beetz, 2016) is widely used as W3C recommendation because it considers semantic relationships between entities within RDF graphs. Geospatial data can also be queried with a SPARQL-based query type named GeoSPARQL (buildingSmart Regulatory Room, 2017; McGlinn et al., 2019), a query type recommended by the OGC for querying linked geospatial data for the semantic web. Another SPARQL-based query BimSPARQL (Zhang et al., 2018) is developed to support BIM-specific characteristics. BIM data can also be queried with other query languages like BIMQL (Mazairac and Beetz, 2013; Schwabe et al., 2019) queries which is based on SQL to combines query languages and BIM specifications. BIM data can also be queried using QL4BIM (Daum and Borrmann, 2013) which is an advanced query language for building information models proposed to fill the lack of spatial functionality in BIM query languages. SPARQL-based queries have a large use domain, we have showed that it has been used to query semantic data, geospatial RDF-based data, and BIM data. Moreover, it has been used also to query IFC binary data serialized in HDF5 (Krijnen and Beetz, 2018). RKQL (Dimyadi et al., 2016) is another SQL-based query language designed to query regulatory data in the AEC field. It provides a simple specification to help interface system developers or building designers write or maintain high-level scripts that can be easily embedded into the compliant design procedures.

The ACCORD project can utilize a combination of query languages outlined in Table 6 to interrogate data in various scenarios. SPARQL is the primary query language for semantic data in RDF format, while GeoSPARQL is better suited for geospatial RDF data. The query languages can be classified into two main categories depending on the targeted data. The first category contains the query languages that are designed to support semantic data and they are based on SPARQL (including BimSPARQL and GeoSPARQL), while the second category, based on SQL, contains query languages designed for structured non-RDF data (including BIMQL, QL4BIM, and RKQL). Each language has unique features to support more specific data types, such as QL4BIM and GeoSPARQL, both are used for building data with geospatial characteristics, however, GeoSPARQL is a semantic query language designed to support RDF data and QL4BIM is designed for BIM-structure data.

Table 6: Classification of query languages by category

Category	Query language	Data characteristics
Designed semantic data	SPARQL	RDF

	GeoSPARQL	RDF, Geospatial
	BimSPARQL	RDF, BIM-structure
Designed for structured non-RDF data	BIMQL	BIM-structure
	QL4BIM	Geospatial, BIM-structure
	RKQL	Regulatory data

3.4 Summary

This section focuses on query languages that can be utilized for data retrieval and querying purposes. These query languages can be employed at different stages of the project, either for selecting data or for verifying the relevance of obtained results to the expected ones. They may be integral to WP4 tasks and play a crucial role in the testing phases. Additionally, querying will be essential during the prototype solutions demonstrations in T5.1 (Automated BIM-based Building Permit and Environmental Compliance – Finland & Estonia Demo), T5.2 (Automated Checking for Land Use Permitting, Green Building Certification and Architectural Design Compliance of Industrialized Timber Housing), T5.3 (Automatic Checking of Structural Integrity of Steel Modular House Components – UK Demo), and T5.4 (Automated Checking of Compliance with Urban Regulations - Spain Demo) to retrieve and analyse the results and prototypes' output, and to demonstrate those results. These query languages encompass various types of data that may arise within the AEC field, including geospatial data and semantic data. So, they can support the tasks associated with ACC.

4. Rule Languages for ACC

Automatic compliance checking heavily relies on checking rules. Rule-based checking system usually defines a general pattern or rules to attend two goals: attending some conclusions based on existing relationships or validate the data according to the data model.

In the literature, Authors in (Eastman et al., 2009; Nawari, 2018) have suggested that the rule-based systems can apply rules with 4 conclusions: “pass”, “fail” or “warning”, or “unknown”. They have also decomposed the automated rule verification process into four phases:

1. Rule interpretation and logical structuring of rules for their application.
2. Building model preparation, where the necessary information required for checking is prepared.
3. The rule execution phase, which carries out the checking.
4. The reporting of the compliance-checking results

Authors in (Solihin et al., 2018) have made a review of language-based rule checking system for building information. In this review, they classified the domain specific languages into two categories. The first one contains what considered as internal languages that uses existing language as the host language, then extend it and adapt it to satisfy the domain specifications, for example: Semantic Web Rule Language (SWRL) (Beach et al., 2015; Dimyadi et al., 2016; Fahad et al., 2017; Farias et al., 2014; Godager, 2018), SPARQL Inferencing Notation (SPIN)[19] (Jakob Beetz and Madhumitha Senthilvel, 2022), RuleML[20], etc. The external domain specific languages are more concise in defining the BIM-based rules, such as Building Environment Rule and Analysis Language (BERA) (Lee, 2011; Nawari, 2019) and BIM Rule Language (BIMRL) (Solihin and Eastman, 2016).

This section is dedicated to the rules used for the automatic compliance checking that could be used for automatic building permit validation. We represent in the first sub-section the different classifications for the rules from different perspectives. Sub-section two contains the different compliance checking types and methods. Then, in the third sub-section we list the rules, their descriptions, and the use cases in the building field. We discuss in the last sub-section the possible ways to apply those rules in the building permitting process, also which rule class or approach or combination of more than one can satisfy this goal.

4.1 Rules classification

We present in this sub-section the different rules classifications; many works have classified the rules according to different perspectives. The rules were classified in (Solihin and Eastman, 2016) according to the approach that uses the rule and its aim. There are four different approaches:

- The 1st approach (semantic rules to check the data model): the objective of this approach is to validate the data model. Data will be translated into RDF format to enable the semantic rule checking feature. For example, SHApes Constraint Language (SHACL) and Shape Expressions (ShEx) are two widely used rules that provide validation and checking of data models in ontologies.
- The 2nd approach (First Order Logic (FOL) rules, used for inference or to predict some conclusion: “pass”, “fail” or “warning”, or “unknown”): using rule with the pattern “IF <condition> THEN <action>”. A rule engine (reasoner) will use the defined rules to perform inference from the building knowledge graph. Several rule languages can be used to express FOL rules, such as SWRL, Prolog, and Drools.
- The 3rd approach (rules hard coded into the system): the objective of this approach is to deliver a system with parametrized rules. It includes systems with hard-coded rules like Solibri and FORNAX.
- The 4th category of approaches is the language-driven approach. This approach relies on the domain-specific language such as BERA. Those approaches are usually driven by the rule experts which make them more transparent. Numeric Data of Building Circulation (NDBC) (Nawari, 2019) is a good example of applications that uses BERA language to analyse the indoor space.

Table 7 displays the corresponding rules that align with each approach along with their respective objectives:

Table 7: Rules classification by approach

Approach	Aim	Rules example
1st approach	Data model checking	SHACL, ShEx

[19] https://www.w3.org/Submission/spin-overview/
[20] http://xml.coverpages.org/ruleML.html

2nd approach	Inference (reasoning on the ontologies to add new facts or to predict some conclusions)	SWRL, Drools, Jena rules, Jess, Prolog
3rd approach	Easily parametrized rules	hard-coded rules in Solibri and FORNAX
4th category	Expert-driven rules, transparent rules	BERA

Authors in (Solihin and Eastman, 2016) have also classified the rules according to their requirements, they distinguished three levels of requirement of each rule class:

- Class-1 rule (rules that require a single or small number of explicit data): include rules that use a simple query function that checks for the existence of a property or a classification.
- Class-2 rule (Rules that require simple derived attribute values): include rules that check the relation among the individuals, for example the components' location; "The discharge pipe shall not be located in places where it can cause health and safety hazards such as locating the discharge pipe above any portable water storage tank and electrical transformer/switchgear".
- Class-3 rule (Rules that require extended data structure): include rules with a detailed cases and more complicated level of checking. Those rules can be divided in sub rules, for example "Doors, when fully opened, and handrails shall not reduce the required means of egress width by more than 7 inches (178 mm). Doors in any position …".

Table 8 shows the required inputs for each class of rules:

Table 8: Rules classification by class

Rule class	Requirements
1st class	Data properties, concepts
2nd class	Object properties, data properties, concepts
3rd class	Composition of the 2nd class rules (a combination of if-else blocks of rules)

There is also another classification according to the rule strategy (Pauwels and Zhang, 2015):

- Strategy 1, Hard-coded rule checking after querying for information: which is implemented by applications like Solibri. In this approach, RDF and OWL are typically not used for storing rule information.
- Strategy 2, Rule-checking by querying: the rule information is formalised directly into SPARQL queries. where the rule information is directly formalized into SPARQL queries. Unlike the first strategy, only the building model is represented in RDF and not the rule information.
- Strategy 3, Semantic rule checking with dedicated rule languages: relies on specific semantic web rule languages such as SWRL, Jess, and N3Logic.

Table 9 summarizes the strategies with the used building model and the type of rules:

Table 9: Rules classification by strategy

Strategy	Building model representation	Rule information representation
Strategy 1	Not stored in RDF/OWL	Hard-coded rules in systems like Solbri and Fornax
Strategy 2	RDF or/and OWL	SPARQL
Strategy 2	RDF or/and OWL	Semantic web rules like SWRL, Drools, Jena rules, Jess, and Prolog.

4.2 Rule languages

We describe in this sub-section the list of existing rule languages used for reasoning or to validate the data shape according to the data model. Table 10 describes 23 rule languages.

Table 10: Rule languages

Rule language	Description	Applications in ACC	Reference
BERA	Building Environment Rule and Analysis Language (BERA) is a domain-specific language, deals with building information models in an intuitive way in order to ensure the quality of design and assess the design programming requirements using user-defined rules in the early design phases	(Nawari, 2019): Automating the indoor spatial validation process with BERA to validate the NDBC.	(Lee et al., 2015)
BIMRL	BIM Rule Language (BIMRL) is a complete environment for data, rule definition, and rule execution.	(Solihin and Eastman, 2016): BIMRL can be used to validate the data.	(Solihin and Eastman, 2016)
Datalog	Datalog is a declarative logic programming language. While it is syntactically a subset of Prolog, Datalog generally uses a bottom-up rather than top-down evaluation model. This difference yields significantly different behaviour and properties from Prolog. Datalog also supports negation in the rules.	(Zadeh et al., 2019): Datalog is used as schema-mapping language to represent the conjunctive queries in BIM-CITYGML Data Integration (BCDI).	website
Drools	Drools is a business-rule management system with forward and backward chaining. A forward-chaining rule system starts with a fact and reacts to its changes, being a data-driven system. Contrarily, a backward-chaining rule system starts with a conclusion that needs to satisfy and continues the process until the initial conclusion or its sub-goals are satisfied, being a goal-driven system.	(Shen et al., 2022): Initially, the safety risk rules were expressed in SWRL language and then transformed into Drools rule language to perform the reasoning. (Schwabe et al., 2019): Drools rules are written by the domain experts then the system applies them to check the geometry compliance.	website

Rule language	Description	Applications in ACC	Reference
IfcConstraint	The purpose of an IfcConstraint is to establish a boundary condition, constraint, or limiting value that can be imposed on either an object or the property value. This constraint can be linked to any subtype of IfcObjectDefinition or IfcPropertyDefinition using the IfcRelAssociatesConstraint relationship to indicate a constraint defined by the system. Alternatively, it can be linked to an IfcResourceObjectSelect, such as IfcPropertySingleValue, through IfcResourceConstraintRelationship to indicate a constraint defined by the user.	(buildingSmart Regulatory Room, 2017): Review and compare IfcConstraint with other 12 languages on one clause from the Korean Building Code.	website
Jena rules	Jena rules are defined by a Java Rule object with a list of body terms (premises), a list of head terms (conclusions) and an optional name and optional direction. Jena defines a set of built-in predicates for comparison and mathematic operations (such as: lessThan, sum, isLiteral, etc). For example: *rule* := *term_1*, … *term_n* -> *hterm_1*, … *hterm_n* In this example, term_i are the body predicates and hterm_i are the head predicates. Every term is a triplet or a built-in predicate.	To the best of our knowledge, it has not been applied yet.	website
Jess rules	Java Expert System Shell (Jess) rules are similar to an IF-THEN statement in a procedural language like Java or C. An IF-THEN rule can be expressed in a mixture of natural language and computer language as follows: IF certain conditions are true THEN execute the following actions	To the best of our knowledge, it has not been applied yet.	website
KBim Code	KBimCode or Korea BIM (KBIM) is a specialized language designed to represent Korean Building codes in an explicit and machine-readable format using a scripting language. This approach separates the process of creating rules from the traditional dependence on rule-checking software.	(JAEYEOL SONG et al., 2019): Translate KBimCode into an executable code of specific rule checking software, named KBimAssess. (Hayan Kim et al., 2018): KBimCode was defined as a neutral language that is composed of translated building regulations as a computer-executable ruleset file. (Kim et al., 2019): Represent KBIMCode with a visual language: KBVL	(JAEYEOL SONG et al., 2019)

Rule language	Description	Applications in ACC	Reference
		(Kim and Lee, 2016): KBimCode is used to represent a computer-readable form of Korean Building Code sentences. (Lee et al., 2016): Using KBimCode for verifying Building Act compliance.	
KBVL	KBim Visual Language (KBVL) is an approach that employs visual symbols to create a machine-readable building code. This visual language approach is advantageous in that it is easy to use for those without programming expertise, thanks to its use of visual symbols, and is highly intuitive due to its user-defined level of visualization.	(Kim et al., 2019): This study shows how KBVL analyses sentences to identify the characteristics of building regulations. The components of sentences are then visualized according to their grammatical and functional properties, and the relationship between visual symbols is established. Finally, the authors demonstrate KBVL using real building regulations.	(Kim et al., 2019)
LegalRuleML	LegalRuleMl is based on the open standard RuleML to represent the logical content and semantics of the document. it was proposed to extend RuleML with formal features specific to legal norms, guidelines, policies and reasoning. It uses Extensible Markup Language (XML) to express rules in the legal domain and to manage legal resources.	(McGibbney and Kumar, 2013): Akoma Ntoso used LegalRuleML to model the content of legal norms e.g. obligations, rights, permissions, etc. in order to permit legal reasoning. (Dimyadi et al., 2016): They used LegalRuleML to model their regulation ontology.	website
LinkML	Linked Data Modeling Language (LinkML) is a flexible modelling language that allows to define schemas in YAML to describe the structure of data. Additionally, it is a framework for validating data in a variety of formats (JavaScript Object Notation (JSON), RDF, Tabulation-Separated Values (TSV)), with generators for compiling LinkML schemas to other frameworks.	To the best of our knowledge, it has not been applied yet.	website
LKIF rules	The Legal Knowledge Interchange Format (LKIF) rule is a semantic web-based language for representing legal knowledge in order to support modelling of legal domains and to facilitate interchange between legal knowledge-based systems. It is a combination between OWL and SWRL.	(Dimyadi and Amor, 2013): LKIF was used as legal data exchange formats. (Dimyadi et al., 2016): LKIF was used to standardize Akoma Ntoso to focus on the semantics and logical content.	website (Gordon, 2010)
N3Logic	Notation 3 Logic is a logic that allows rules to be expressed in a Web environment. It uses N3 syntax and extends RDF with N3 syntax and a	(Pauwels et al., 2011). Express the rules in N3Logic.	(Berners-Lee et al., 2008)

Rule language	Description	Applications in ACC	Reference
	vocabulary of new predicates, which can be used to talk about the provenance of information, contents of documents on the Web, and provide a variety of useful functionality such as string, cryptographic, and mathematic functions.		
Prolog	Prolog or PROgramming in LOGic is a descriptive language which used to solve problems that involve objects and relationships between objects. It has been used for various applications including natural language understanding and reasoning. Unlike conventional programming languages, Prolog focuses more on describing facts and relationships rather than defining a sequence of steps to solve a problem.	(Xue and Zhang, 2022) : This approach used B-Prolog – The high-performance implementation of ISO-Prolog which also supports many data structures. (link) (Di Martino et al., 2019): They used a Prolog based inference engine. (buildingSmart Regulatory Room, 2017): Review and compare Prolog with other 12 languages on one clause from the Korean Building Code. (Eilif Hjelseth and Beidi Li, 2021): ASP4BIM (a declarative spatial reasoner) enhances previous declarative spatial reasoning systems like λProlog(QS) (Li et al., 2019) by providing non-monotonic reasoning capabilities about real-world buildings with numerous, complex semantic and geometric details. (Zheng et al., 2022): Showed the difficulty in dealing with implicit information (e.g. implicitly defined quantity information like the number of safe exits contained in a room)	(Clocksin and Mellish, 2003; Körner et al., 2022) website
RASE	The foundation for RASE is using mark-up based on four operators: requirement (R), applicability (A), selection (S) and exceptions (E). Briefly, applicability, selection and exception define the scope of the decision and the requirements define the decision itself. This is a semantic-based concept to transform normative documents into a well-defined rule which can be implemented into BIM/IFC-based model-checking software.	(Beach et al., 2015): Proposed expansions to the RASE terminology (E A de Mendonça et al., 2020): The Brazilian accessibility code was represented using RASE methodologies to define a set of rules in Solibri Model Checker.	(Eilif Hjelseth and Nick Nisbet, 2011, 2010)
RIF	The Rule Interchange Format (RIF) is a W3C standard for exchanging rules. RIF has several “flavours” addressing different features of rule	To the best of our knowledge, it has not been applied yet.	(Kifer, 2008)

Rule language	Description	Applications in ACC	Reference
	languages. It can be serialised in RDF or in a more compact format.		
Rule Tables	The use of rule tables is prevalent for informally documenting knowledge and anticipated outcomes. Spreadsheets have been employed to enable domain participants to access and modify these rules easily. The rule format can be either row or column based.	(buildingSmart Regulatory Room, 2017): Review and compare Rule Tables with other 12 languages on one clause from the Korean Building Code.	(buildingSmart Regulatory Room, 2017)
RuleML	Rule Markup Language was developed to express both forward and backward rules in XML for deduction, rewriting, and further inferential-transformational tasks.	To the best of our knowledge, it has not been applied yet.	website presentation
SHACL	SHApes Constraint Language (SHACL) is a language for describing and validating RDF graphs. This validation can ensure the conformance of RDF data to a defined schema.	(Robaldo, 2021): First attempt to investigate how to serialise reified I/O formula modelling obligations as SHACL shapes and reified I/O formula modelling constitutive rules as SHACL rules. (Stolk and McGlinn, 2020): A method for validating ifcOWL models using SHACL. (Cao et al., 2022): The validation rules are coded in SHACL.	website SHACL Rules
ShEx	Shape Expressions (ShEx) is a structural schema language for RDF graphs. It allows to define datatype constraints and to describe profiles of data.	To the best of our knowledge, it has not been applied yet.	website
SPIN	SPARQL Inferencing Notation (SPIN) is a SPARQL-based rule and constraint language for the Semantic Web. SPIN is also a mechanism to represent reusable SPARQL queries as templates and to define new SPARQL functions with a web-friendly syntax. It is considered as the predecessor of SHACL.	(Jakob Beetz and Madhumitha Senthilvel, 2022): They used Spin alongside SHACL to validate data.	website
SWRL	Semantic Web Rule Language (SWRL) is a combination of the OWL DL and OWL Lite sublanguages with the Binary Dialog RuleML sublanguage of the RuleML. This extends OWL axioms with Horn-like rules (rules which represent implications between an antecedent (body) and consequent (head)).	(Zheng et al., 2022): Used for ontology semantic enrichment (to derive implicit spatial relationships). (Beach et al., 2015): The use of SWRL based-rules during the compliance checking process. (buildingSmart Regulatory Room, 2017): Review and compare SWRL with other 12	website

Rule language	Description	Applications in ACC	Reference
		languages on one clause from the Korean Building Code. (Bus et al., 2018): Compliance checking based on SWRL semantic rules. (Dimyadi et al., 2016): Modelling regulatory knowledge using semantic web technologies: RDF + OWL + SPARQL + SWRL. (Fahad et al., 2016): Compare MVDXML and SWRL technologies for the model instance verification and conformance checking of IFC models. (Fahad et al., 2017): Validation of IFC Models with SWRL. (Farias et al., 2014): Define SWRL rules and show the benefits of applying SWRL rules to handle IFC files. (Farias et al., 2018): Reasoning with SWRL rules. (Fortineau et al., 2019): Using SWRL rules for BIM based information systems checking. (Godager, 2018): Using SWRL to enrich an OWL version of IFC while facilitating the use of reasoning engines. (Hu et al., 2021): SWRL is used to enrich the OWL version for IFC and create the semantic rule checking environment. (Lu et al., 2015): Safety checking constraints are represented with SWRL rules (P. Pauwels et al., 2017): Using SWRL rules and SPARQL queries to convert IFC geometry into alternative geometric representations. Also, SWRL rules are used to generate a simplified version	

Rule language	Description	Applications in ACC	Reference
		of the ifcOWL ontology. It is also shown that the information captured in an mvdXML file can also be captured using SWRL rules. (Pieter Pauwels et al., 2017a): Rule checking over SWRL rules with a semantic graph database. (Pauwels and Zhang, 2015): Semantic rule checking with dedicated rule languages that relies on dedicated semantic web rule like SWRL. (Zhong et al., 2012): Automated construction quality compliance checking with SWRL rules.	
VCCL	Visual Compliance Checking Language (VCCL) is a programming language specifically designed for creating verification and checking procedures that conform to established standards or guidelines. By utilizing digital building information, VCCL can perform compliance checks in a fully or partially automated manner.	(Preidel and Borrmann, 2015): VCCL methods for Automated Code Compliance Checking. (Preidel and Borrmann, 2016): This study presents the features and functionalities of VCCL in detail and shows its application in a number of case studies for code compliance checking. (Preidel et al., 2017): Using VCCL for data analysis and processing tasks in the context of Building Information Modelling.	(Preidel and Borrmann, 2017)

4.3 A Comparison of Rule Languages

This part of our study involves a comparison of the rule languages presented in this section using two main criteria. Firstly, we assess their support for basic logical operators, which include conjunction (AND), disjunction (OR), and negation (NOT). Secondly, we evaluate the languages' support for Built-In predicates, including comparison predicates (equal, lessThan, greaterThanOrEqual, etc.), mathematical predicates (add, divide, sin, etc.), string predicates (stringLength, lowerCase, startsWith, etc.), temporal predicates (dayTimeDuration, addDayTimeDurations, etc.), and list predicates (member, length, empty, first, etc).

Table 11 shows that all the rule languages support the conjunction operator, and most of them support disjunction and negation, although their syntax may vary between languages. However, only a few rule languages have Built-In predicates that enhance their expressivity, such as Drools, Jena rules, Prolog rules, and SWRL.

Table 11: Comparison of rule languages

Rule Language	Supported logical operators			Built-In Predicates				
	Conj.	Disj.	Neg.	Comparisons	Math	Strings	Date/time	Lists
BERA	X	X	X					
BIMRL	X	X	X					
Datalog	X		X					
Drools	X	X	X	X	X	X	X	X
IfcConstraint	X	X	X					
Jena rules	X	X	X	X	X	X	X	X
Jess rules	X	X	X	X	X	X	X	X
KBim Code	X	X	X					
KBVL	X	X	X					
LegalRuleML	X	X	X	X			X	
LinkML	X	X						
LKIF rules	X	X	X					
N3Logic	X	X	X	X	X	X	X	X
Prolog	X	X	X	X	X	X	X	X
RASE	X	X	X					
RIF	X	X	X	X		X	X	X
Rule Tables	X	X		X				
RuleML	X	X	X	X	X	X	X	X
SHACL	X	X	X	X	X	X	X	X
ShEx	X	X	X					
SPIN	X	X	X	X	X	X	X	X
SWRL	X	X	X	X	X	X	X	X
VCCL	X				X			

4.4 Summary

We have presented in this section the rule-based checking system concept and the existing rule classifications according to different perspectives. The rules could be classified depending on the approach that uses the rule and its aim, on the rule requirements, or on the rule strategy. Those classifications are compatible with each other, it means that the same rule can be classified according to the three existing classifications. However, we can notice some similarities between the approach-based classification and the strategy-based classification. Both classifications have considered the hard-coded rules as an independent class. We also notice that the strategy-based classification considers the semantic rule checking as a single classification, in the other hand the approach-based classification distinguishes between the semantic rules to check the data model and the inference rules and consider them to be two separate classes. The strategy-based classification adds a new class for the rule-checking by querying. Beside the approach-based classification that also add a new category for the rules with domain-specific language.

We have also presented the existing classifications for the automated compliance checking (ACC) systems. They are also classified according to different criteria: the checking type of the used method, the system characteristics, and the existing methods for ACC. The first two classifications focused mainly on the four ACC methods: validation checking, model content checking, smart object checking, and design option checking. The last classification went through 6 detailed methods for ACC.

In sub-sections 4.2 and 4.3, we have presented and compared between the existing rule languages for the ACC systems. Some of those rules were used by some works and have proven their efficiency in the AEC field. Some rule languages were designed to fit some type of checking such as SHACL and ShEx for data shape validation, notice that SHACL is a W3C recommendation (for more comparing criteria between SHACL and ShEx: website). Some of those rules are Horn clause to express first order logic expressions such as SWRL. Some of the rules are rich with built-in predicates to express some mathematic relations and comparisons like SWRL and Metalog. This list of rule languages contains rules from both categories: those considered as internal languages because they have been built upon an existing language such as SWRL which is a combination of OWL-DL and RuleML. An example of an external rule language is BERA that is more concise and specialized for the building field.

Nonetheless, we have noticed a shortage of rules that are easily accessible to regulation experts who are the primary users for ACC systems. The only rule languages available for this purpose are RASE and VCC, but they are hindered by limited expressivity, such as the absence of built-in predicates.

The ACCORD project's ACC task will heavily rely on reasoning and validation rules, which are the focus of T2.3 (Machine-executable Regulations) and T2.5 (Design and Implementation of Rule Formalisation Tool). T2.3 can benefit from the literature review presented in this section, which primarily focuses on rule classification. This literature review will aid in defining the project's methodology by selecting and adapting existing approaches discussed here. Furthermore, this section will also assist T2.5 in choosing the most suitable rule languages based on the specific requirements of each task and the intended data set or operation (reasoning or validation).

5. Standards and Recommendations for ACC

In this section, we will focus on the standards pertaining to building permit compliance checking, with the objective of gathering only the recommendations and standards applicable to the ACCORD context. These standards can serve as a valuable source of inspiration for the construction of AEC3PO. Additionally, they can be used to design and represent compliance rules effectively. An initial list of standards for this Section was drawn from D1.1, these were all analysed and references from these followed to expand the standards list.

This section is divided into three subsections, the first one defines the scope to select the standards. The second sub-section presents a compilation of standards that are classified based on their intended functionality. For instance, standards that establish a lexicon are grouped under the terminology category, while those that facilitate the query, and the retrieval of data are placed within the query languages standards category. The concluding part of this section shows the links between this section and the other tasks of the project.

5.1 Scope

In the AEC industry, several standards and recommendations aim to achieve different levels of interoperability in systems, ranging from syntactic standards to governance, policies, legal, and regulatory standards. In this deliverable, we will concentrate on data-related standards such as those that provide syntactic rules and semantics to represent data in a standardized way. Policy and regulatory standards are addressed in the deliverable D1.1 “Landscape Review Report”. Standards are categorized in this deliverable based on their functionality, including aspects like data model, terminology, and syntax.

Standards play a crucial role in various stages of the ACCORD project. The syntactic standards will be applied to represent the data. In addition, data models and terminologies will aid in the development, enrichment, and population of the AEC3PO ontology.

This section includes standards that falls within semantic data solutions approach, including standards from W3C, the ISO, the German Institute for Standardization (DIN), OGC, the Organization for the Advancement of Structured Information Standards (OASIS), the European Committee for Standardization (CEN), buildingSMART International, the National Institute of Standards and Technology (NIST), the Dutch Standardization Organization (NEN), and the European Commission's Joint Research Centre (JRC).

This section will not redefine the standards discussed in the previous sections, such as ontologies, query languages, and rule languages. Instead, it will reference these standards and focus on introducing new ones that have not been previously mentioned.

5.2 Representative standards

In this section, we classify 88 standards that are relevant to our scope into 12 groups based on their characteristics and considering the possible needs for the development of ACC solutions within the ACCORD project:

1. Query languages: encompass the languages used to retrieve RDF data.
2. Ontology languages: contain languages used to formulate the ontology.

3. Schematization languages: contains languages that schematize the data.
4. Rule languages: used to formulate the rules.
5. Validation languages: used to validate the data.
6. Data formats: define the syntax of representing and exchanging data in a specific format.
7. 3D modelling formats: define the syntax of representing and exchanging data in a 3D format.
8. The data models: define the structure of data.
9. Unified Modeling Language (UML) data models: contains data model presented with an UML diagram.
10. Terminology and vocabulary: consist of a collection of vocabularies that describe various aspects related to the building.
11. Taxonomy: contains a classification of building terms within the AEC domain.
12. APIs: contain a set of specifications that define how software components should communicate with each other in a standardized way.

5.2.1 Query languages

Once the built data is transformed into an RDF graph, a query language is required to retrieve and analyse the desired information. SPARQL is a recommended query language by W3C for querying RDF data. Additionally, for geographic RDF data, OGC has recommended GeoSPARQL for geospatial data on the Semantic Web.

5.2.2 Ontology languages

The ontology language standards (see Table 12) provide guidelines and specifications for creating a machine-readable format to represent and share knowledge. The purpose of these standards is to assist in the development of formal and explicit descriptions of concepts and their relationships within a particular domain, known as ontologies.

Table 12: Ontology languages standards

Standard Name	Full name	Description	Related standards	Organisation	Open access	Year	Web Link
OWL	Web Ontology Language	Ontology Web Language - Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things	RDF	OWL Working Group in W3C	Yes	2004	website

5.2.1 Rule languages

The rule language standards that include LegalRuleML, RIF, and RuleML provide a consistent method of expressing logical relationships between data elements and allow for the creation of formal rules to support automated reasoning.

The subsequent significant step in the compliance checking process for building permits is to develop compliance checking rules. Rule languages such as RuleML and LegalRuleML have been standardized by OASIS and are suggested by them for specifying rules. Similarly, RIF has been recommended by W3C to be compatible with RDF and OWL for this purpose (see section 4.2 "Rule

Languages" for more information). In the context of AEC, RASE is well established and may be considered essential as a starting point for further investigation in the ACCORD project.

5.2.2 Validation languages

The standards listed in Table 13 pertain to validation language standards, which consist of a set of specifications that outline the syntax and semantics for validating data against a predefined set of rules. These standards offer an automated and uniform approach for verifying the accuracy and completeness of data across various applications.

Table 13: Validation languages standards

Standard Name	Full name	Description	Related standards	Organisation	Open access	Year	Web Link
XML Schema	XML Schema Definition Language	XML Schema describes the structure of an XML document. XML Schema language is also referred to as XML Schema Definition.	XML	W3C XML Schema Working Group	Yes	2012	website
JSON Schema	JSON Schema Specifications	JSON Schema is a declarative language that enables annotation and validation of JSON documents. JSON Schema Core defines the basic foundation of JSON Schema, and JSON Schema Validation defines the validation keywords of JSON Schema	JSON	OpenJS Foundation	Yes	2022	website

Validation standards in Table 13 can be utilized to ensure that the modelled data is accurate. SHACL (from Section 4) is one such standard, which is well-suited for validating semantic data. Additionally, the structure of an XML document can be checked according to XML Schema Definition standard.

5.2.3 Schematization languages

This category contains conceptual schema languages that support AEC domain specifications (see Table 14).

Table 14: Schematization languages

Standard Name	Full name	Description	Related standards	Organisation	Open access	Year	Web Link
EXPRESS	also described in ISO 10303-11:2004 Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual	The EXPRESS Definition Language for IFC Development: is a conceptual schema language which provides for the specification of classes belonging to a defined domain, the information or attributes pertaining to those classes (colour, size, shape etc.) and the constraints on those classes (unique, exclusive etc.). EXPRESS is formalized in the ISO Standard for the Exchange of Product model STEP (ISO 10303).	IFC	ISO TC 184	Yes	2004	(Farias et al., 2014; Godager, 2018; Hu et al., 2021; Pauwels et al., 2011; "The EXPRESS Definition Language for IFC Development," 2019) website
COBie	Construction-Operations Building information exchange (COBie) standard data	The Construction-Operations Building information exchange (COBie) standard defines information for assets that are delivered as part of a facility construction project and is used to document the data for the BIM process	space, document, component, type, attribute	The U.S. National Institute of Building Sciences (NIBS) and the UK government's BIM Task Group	Yes	2008	website

5.2.4 Data representation formats

The data representation formats define the syntax of representing and exchanging data in a particular format. The standards displayed in Table 15 establish the rules for creating and interpreting data elements and provide a uniform way of representing data across various systems and applications. Syntactic standards for data formats are vital to ensure interoperability between different systems and are essential in facilitating data exchange and integration.

Table 15: Data formats standards

Standard Name	Full name	Description	Related standards	Organisation	Open access	Year	Web Link
JSON (ISO/IEC 21778:2017)	ISO/IEC 21778:2017 Information technology — The JSON data interchange syntax	Javascript Object Notation		the ISO and the International Electrotechnical Commission (IEC)	No	2017	website
RDF	Resource Description Framework	Resource Description framework - standard model for data interchange on the Web.	RDFS	W3C	Yes	1999	website
STEP	ISO 10303, it is known informally as STEP: Standard for the Exchange of Product model data	STEP data format, commonly used to exchange IFC data. It is defined in ISO 10303-242:2014 Industrial automation systems and integration — Product data representation and exchange — Part 242: Application protocol: Managed model-based 3D engineering.	IFC, EXPRESS	ISO TC 184	No	2014	website
XML	Extensible Markup Language	Extensible Markuplanguage	XML Schema	W3C	Yes	1998	website

Table 15 presents the foundational standards such as XML and JSON, which are syntax standards for data formats that have been recommended by W3C and ISO. There are also other standards for data formats, such as STEP which is commonly used to exchange IFC data. To represent semantic data as triplets and construct a graph of data, RDF is designed as a triplet-based representation, which is mandatory for representing relations between data.

5.2.5 Standards for 3D modelling

The standards for 3D modelling specify the guidelines and specifications for defining the syntax and structure of representing and exchanging 3D models. These standards establish the rules for creating and interpreting 3D models and provide a uniform way of representing 3D models across different systems and applications. Syntactic standards for 3D modelling in Table 16 are crucial to ensure interoperability between different systems that use 3D building models and have an integral role in facilitating data exchange and integration.

Table 16: 3D modeling standards

Standard Name	Description	Characteristics	Organisation	Open access	Year	Web Link
3DTiles	3D Tiles is designed for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds	3D, geospatial	OGC	Yes	2022	website
Indexed 3D Scene Layers (I3S)	A container for arbitrarily large amounts of heterogeneously distributed 3D geographic data.	3D geographic format	OGC	Yes	2022	website

3D modelling standards in Table 16, including Indexed 3D Scene Layers (I3S) and 3DTiles are designed for rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD (Computer-Aided Design).

5.2.6 Data models

Data model standards define the structure of data for the building domain or application. These standards (presented in Table 17) provide a standardized way of representing and organizing data elements and relationships in the AEC field and other fields.

Table 17: Data model standards

Standard Name	Full name	Description	Supported format	Related standards	Organisation	Open access	Year	Web Link
ADMS	Asset Description Metadata Schema	ADMS (Asset Description Metadata Schema) is used to describe semantic assets as highly reusable metadata	XML	XML Schema	W3C Working Group	Yes	2013	website
BCF	The BIM Collaboration Format	BCF File Based Transmission	XML, OWL	XML Schema, BCF	The Open CDE workgroup	Yes	2020	website

Standard Name	Full name	Description	Supported format	Related standards	Organisation	Open access	Year	Web Link
CEN EN 17632	Building information modelling (BIM) - Semantic modelling and linking (SML) - Part 1: Generic modelling patterns	Semantic modelling and linking (SML): addresses syntactic and semantic interoperability for information describing assets going through their life cycle in the built environment	XML	XML Schema	The European Committee for Standardization (CEN)	No	2018	website
CEN WI 442018		Exchange structure for product data templates and product data based on ifcXML	XML	XML Schema, IFC	CEN	/	/	(Henttinen, 2020)
CEN WI 442033		Exchange structure for product data templates and product data based on ifcXML - Part 2 Requirements and configurable products	XML	XML Schema, IFC	CEN	/	/	(Henttinen, 2020)
CityGML		The CityGML standard defines a conceptual model and exchange format for the representation, storage and exchange of virtual 3D city models.	XML	XML Schema, GML	OGC	Yes	2021	website
CityJSON		JSON Encoding for CityGML. CityJSON is a JSON-based encoding for storing 3D city models, also called digital maquettes or digital twins (longer explanation)	JSON	JSON (ISO/IEC 21778:2017)	Delft University of Technology in the Netherlands	Yes	2016	website

Standard Name	Full name	Description	Supported format	Related standards	Organisation	Open access	Year	Web Link
eCOB	eCOB BIM object creation standard	The eCOB standard for the Creation of BIM Objects is an instrument for generating generic or industrial BIM objects with an information structure, facilitating interoperability between BIM programs throughout the entire life cycle of the construction. eCOB® is based on IFC, the European Harmonized regulatory context and the National regulations applicable to construction projects in a specific country. At this time it is adapted to Spanish regulations (Technical Building Code, EHE, etc).	XML	XML Schema, IFC	ITeC foundation	Yes	2021	website
ETIM	ETIM, the international classification standard for technical products (from German: ElektroTechnisches InformationsModell)	European Technical Information Model (from German: ElektroTechnisches InformationsModell) is an open standard for the unambiguous grouping and specification of products in the technical sector through a uniform classification system.	XML	XML Schema	ETIM International	Yes	2019	website
GML	ISO 19136:2007 Geographic information — Geography Markup Language (GML)	Geography Markup Language		XML Schema	OGC	Yes	2012	website website
Haystack		Project Haystack is an open source suite of technologies for modeling IoT data	JSON	JSON (ISO/IEC 21778:2017)	Haystack connect organization	Yes	2019	website

Standard Name	Full name	Description	Supported format	Related standards	Organisation	Open access	Year	Web Link
IFC	Industry Foundation Classes	Standardised Digital Description of the Built Environment	EXPRESS, OWL, JSON, XML, STEP, HDF5, RDF	EXPRESS	buildingSMART International	Yes	1999	website
IndoorGML	OGC Standard for Indoor Spatial Information	IndoorGML is an OGC standard for an open data model and XML schema for indoor spatial information. It aims to provide a common framework of representation and exchange of indoor spatial information	GML	XML Schema, GML	OGC	Yes	2019	website
InfraGML	Land and Infrastructure Conceptual Model Standard GML	Encoding of LandInfra in GML	GML	XML Schema, GML	OGC	Yes	2018	website
ISO 10303	ISO 10303-21:2016 Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure	Specifies an exchange format that allows product data described in the EXPRESS language to be transferred from one computer system to another	EXPRESS	EXPRESS	ISO TC 184	No	2016	website
ISO 16739	ISO 16739-1:2018 IFC for data sharing in the construction and facility management industries — Part 1: Data schema	Specifies a conceptual data schema and an exchange file format for Building Information Model (BIM) data.	XML	XML Schema, XSD, IFC	ISO TC 59	No	2018	website

Standard Name	Full name	Description	Supported format	Related standards	Organisation	Open access	Year	Web Link
ISO 22057	ISO 22057:2022 Sustainability in buildings and civil engineering works — Data templates for the use of environmental product declarations (EPDs) for construction products in building information modelling (BIM)	Data templates for the use of electronic product dictionaries	XML	XML Schema	ISO TC 59	No	2022	website
LegalDocumentML	OASIS LegalDocumentML (LegalDocML) TC	Use of XML in Legal Documents	XML	XML Schema	the LegalDocML Technical Committee of the Organization for the Advancement of Structured Information Standards (OASIS)	Yes	2008	website
RDFS	RDF Schema	RDF Schema (RDFS) is an extension of the RDF that enables the creation of vocabularies and ontologies for the Semantic Web. RDFS facilitates the definition of classes and properties that govern the structure and relationships of resources within an RDF graph.		RDF	W3C XML Schema Working Group	Yes	2014	website
SKOS	Simple Knowledge Organization System	SKOS is a standard vocabulary and knowledge organization system that is used to describe concepts and their relationships. Its purpose is to provide a way to structure content and data for improved discovery and searchability in the Semantic		RDF	W3C XML Schema Working Group	Yes	2009	website

Standard Name	Full name	Description	Supported format	Related standards	Organisation	Open access	Year	Web Link
		Web. SKOS can be used to develop controlled vocabularies and taxonomies. It is a lightweight and adaptable standard that can be easely integrated with other Semantic Web technologies.						
Xplanung	XPlanung - Planning and Zoning for Germany	A single format and information model for all spatial planning.		XML Schema	The Federal Ministry of the Interior, Building and Community (BMI) in cooperation with the Federal Agency for Cartography and Geodesy (BKG)	Yes	2005	website website
XSD	XML Schema Definition Language	XML Schema Definition Language (XSD) offers facilities for describing the structure and constraining the contents of XML documents, including those which exploit the XML Namespace facility		XML Schema	W3C	Yes	2012	website

The data models presented in Table 17 are based on various schemas such as Extensible Markup Language (XML), JavaScript Object Notation (JSON), and EXPRESS. ISO 16739 and ISO 10303 are among the most useful standards for the project as they respectively specify a conceptual data schema for Building Information Model (BIM) data and an exchange format for product data. Both standards express their data models with EXPRESS. IFC is another important EXPRESS-based data model that is widely used in the building field (Di Martino et al., 2020; Future Insight Group, 2019; Kacfah Emani et al., 2015). CityGML and CityJSON, which are respectively based on XML schema and JSON, define a conceptual model and exchange format for the representation of 3D city models. Similarly, other standards such as Geography Markup Language (GML), IndoorGML, and InfraGML are designed for representing geographic data models.

5.2.7 UML data models

UML data model for building are presented with an UML diagram. The standards in Table 18 provides a graphical representation of different aspects of building system, including its structure, behaviour, and interactions.

Table 18: Standards for UML data models

Standard Name	Full name	Description	Characteristics	Related standards	Organisation	Open access	Year	Web Link
CB-NL	Dutch concept library cb-nl	The CB-NL (Concept Library for the Built Environment) delivers an unambiguous description of built environment concepts	environment		The Dutch Building Information Council (BIR)	Yes	2013	website
CEN PREN 17473	Building information modelling (BIM) - Data templates for construction objects used in the life cycle of any built asset - Data templates based on harmonised technical specifications under the Construction Products Regulation (CPR)	Data templates for construction objects used in the life cycle of any built asset - Data templates based on harmonised technical specifications under the CPR	products, units		CEN	No	2020	website
CEN PREN 17549-1	Building information modelling - Information structure based on EN ISO 16739 1 to exchange data templates and data sheets for construction objects - Part 1: Data templates and configured construction objects	This standard defines a format for exchanging empty product data templates and filled product data templates and therefore fills the missing link between the product data sources from the manufacturers to the construction models of the designers and owners.	products	EN ISO 16739	CEN	No	2022	website

Standard Name	Full name	Description	Characteristics	Related standards	Organisation	Open access	Year	Web Link
CEN WI 442035		Data templates based on European standards and technical specifications			CEN	No	2021	(*NSAI TECHNICAL COMMITTEES NSAI/TC 68 BUILDING INFORMATION MODELLING (BIM)*, 2021)
FSGIM	The Facility Smart Grid Information Model	FSGIM standard is one part of a larger ecosystem of standards that support the development and implementation of a smart electric grid. The FSGIM uses Unified Modeling Language to define key concepts that must map between electricity providers and electricity consumers	electricity		NSIT	No	2016	website
INSPIRE UML model	INSPIRE Consolidated UML Model	Data structure proposed by the INSPIRE European Directive	geographic		JRC	Yes	2013	website
ISO 19152:2012	ISO 19152:2012 Geographic information — Land Administration Domain Model (LADM)	It covers the basic information-related to components of land administration (including those over water and land, and elements above and below the surface of the earth)	geographic		ISO TC 211	No	2012	website
LandInfra	OGC Land and Infrastructure Conceptual Model Standard	Land and Infrastructure Conceptual Model is designed to model land and civil engineering infrastructure facilities.	geographic		OGC	Yes	2016	website

Recommended UML-based data models presented in Table 18 can aid in constructing the schema for the AEC3PO ontology. CEN standards, including CEN PREN 17473 and CEN PREN 17549-1, describe a data template for building products. Geographic data structure models such as

INSPIRE UML model, ISO 19152:2012, Land Administration Domain Model (LADM), and Land and Infrastructure Conceptual Model (LandInfra), were designed to model lands and civil engineering infrastructure facilities.

5.2.8 Terminologies and Vocabularies

Table 19 describes the standards that precis the standards that define a set of terms used within the building domain or application. The standards for building terminology terms definitions to ensure the accuracy of the building description. Standards for building terminology play a crucial role in facilitating communication and understanding in AEC field, ensuring consistency and accuracy of terms, and supporting interoperability between different systems and applications.

Table 19: Standards for building terminology

Standard Name	Full name	Description	Characteristics	Organisation	Open access	Year	Web Link
ISO 16757	ISO 16757-1:2015 Data structures for electronic product catalogues for building services — Part 1: Concepts, architecture and model	Data structures for electronic product catalogues for building services	electronic product	ISO TC 59	No	2015	website
ISO 16818	ISO 16818:2008 Building environment design — Energy efficiency — Terminology	ISO 16818:2008 gives terms and definitions for use in the design of energy efficient buildings.	energy	ISO TC 205	No	2008	website
ISO 18523	ISO 18523-1:2016 Energy performance of buildings — Schedule and condition of building, zone and space usage for energy calculation — Part 1: Non-residential buildings	ISO 18523-1:2016 specifies the formats to present schedule and condition of building, zone and space usage, which is to be referred to as input data of energy calculations for non-residential buildings.	space, schedule	ISO TC 163	No	2016	website

Standard Name	Full name	Description	Characteristics	Organisation	Open access	Year	Web Link
ISO 22496	ISO 22496:2021 Windows and pedestrian doors — Vocabulary	This document specifies general terminology for windows and pedestrian doors.	windows, doors	ISO TC 162	No	2021	website
ISO 22497	ISO 22497:2021 Doors, windows and curtain walling — Curtain walling — Vocabulary	This document provides definitions for terms used in documents, drawings, specifications, etc., when referring to the detailed elements of curtain walling.	walling elements	ISO TC 162	No	2021	website
ISO 6707	ISO 6707-1:2020 Buildings and civil engineering works — Vocabulary — Part 1: General terms	Buildings and civil engineering works: This document contains the terms and definitions of general concepts to establish a vocabulary applicable to buildings and civil engineering works.	building	ISO TC 59	No	2020	website
ISO 8930	ISO 8930:2021 General principles on reliability for structures — Vocabulary	This document establishes the common vocabulary of the principal terms used in the field of reliability of structures.	structures	ISO TC 98	No	2021	website
ISO/CD 7615	ISO/CD 7615-1 Energy performance of building systems — Underfloor air distribution systems — Part 1: Definitions, terminology, technical specifications and symbols	The proposed standard is the first part of a comprehensive series of international standards related to energy performance of underfloor air distribution systems. This standard shall stipulate the definition, terminology, technical specification and symbols related to underfloor air distribution systems.	Energy performance	ISO TC 163	/	Under develo pment	website

Standard Name	Full name	Description	Characteristics	Organisation	Open access	Year	Web Link
ISO/FDIS 17607	ISO/FDIS 17607-1 Steel structures — Execution of structural steelwork — Part 1: General requirements and vocabulary	Determines the general requirements and vocabulary for structural steelwork	steelwork	ISO TC 167	/	Under develo pment	website
NEN 2660:2022	Rules for information modelling of the built environment - Part 1: Conceptual models	Rules for information modelling of the built environment - Part 1: Conceptual models. This standard describes terminology and general rules for a information system for the building field. The standard lays down rules for entities, attributes and models. Part 2: Practical configuration, extension and implementation of NEN 2660-1	architectural, design, terminology, life cycle	The Dutch standardization organization, NEN (Nederlands Normalisatie-instituut)	No	2022	website

The AEC3PO ontology can draw inspiration from standards that define building vocabulary and terminology (see Table 19) when defining concepts. For instance, ISO 22496 defines vocabulary for windows and doors description, while ISO 22497 provides definitions for terms used to describe walling elements. ISO/FDIS 17607 establishes the general requirements and vocabulary for structural steelwork. In terms of general building terminology, the Dutch standardization organization created NEN 2660:2022, while the ISO TC 59 designed ISO 6707 for the same purpose. On the other hand, ISO 16757 provides terms and definitions for use in designing energy-efficient buildings.

5.2.9 Taxonomy

These types of standards organize and classify the building terms in AEC domain. The standards described in Table 20 categorize the building terms according to their characteristics (space, product, material, safety terms, etc.).

Table 20: Standards for building taxonomy

Standard Name	Full name	Description	Characteristics	Organisation	Open access	Year	Web Link
CB-NL	Dutch concept library cb-nl	The CB-NL (Concept Library for the Built Environment) delivers an unambiguous description of built environment concepts	environment	The Dutch Building Information Council (BIR)	Yes	2013	website
CCIC	Construction Classification International Collaboration	It defines a construction classification system	space, construction entities, components	Construction Classification International Collaboration organisation	Yes	2020	website
ECLASS	ECLASS Standard	eClass is the global reference data standard for the classification and unambiguous description of products and services	products, services	ECLASS e.V. association	No	2001	website
ISO 12006	ISO 10303-21:2016 Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure	This standard identifies a set of recommended classification table titles for a range of information object classes according to particular views, e.g. by form or function, supported by definitions. It shows how the object classes classified in each table are related, as a series of systems and sub-systems, e.g. in a building information model.		ISO TC 184	No	2015	website
ISO/TR 22845	ISO/TR 22845:2020 Resilience of buildings and civil engineering works	This document provides an index of typical existing information on concept, disaster risk and countermeasure for resilience of buildings and civil engineering works.	disaster risk and countermeasure	ISO TC 59	No	2020	website
NEN 2699	Investment and operating costs of property - Terminology and classification	This standard gives a classification of working costs and life cycle costing of buildings of real estate, that is: areas, buildings wih sites. Boats and mobile homes are not included	working costs	The Dutch standardization organization NEN	No	2017	website

Standard Name	Full name	Description	Characteristics	Organisation	Open access	Year	Web Link
OMNICLASS	Construction Classification System (known as OmniClass™ or OCCS)	Classificaiton System commonly used in North Amercia. It is used for organizing and retrieving information specifically designed for the construction industry.	construction entities, spaces, elements, products, materials, properties	Construction Specifications Institute (CSI)	No	2001	website
UNICLASS	Unified Construction Classification	UNICLASS - UK Classification Schema for Built Environment Entities: is a unified classification system for the built environment covering all sectors and roles. Uniclass is a way to organize everything required for built environment assets and provide a logical code for each general item, which can be used by anyone to identify and refer to it.	built assets	UK-based Royal Institute of British Architects (RIBA)	Yes	1997	website
UNIFORMAT II	Standard Classification for Building Elements and Related Sitework—UNIFORMAT II	Standard Classification for Building Elements and Related Sitework	elements	American Society for Testing and Materials (ASTM) International	No	2020	website

Standardization organizations and associations have recommended various taxonomy standards (see Table 20) to classify building data, including built assets, products, and elements, such as ECLASS, OMNICLASS, Unified Construction Classification (UNICLASS), and UNIFORMAT II. These taxonomy standards can aid in defining concepts and their hierarchies for the AEC3PO ontology.

5.2.10 APIs

Standardized APIs (Application Programming Interfaces) refer to a set of guidelines and specifications that define how software components should communicate with each other in a standardized way. The building-related APIs in Table 21 are a set of tools for building software applications that provide a consistent and reliable way of integrating different software systems and components. The standardized APIs specify the rules for authentication, security, and error handling. Those APIs are essential for promoting interoperability between different software components and systems in AEC field, enabling developers to build software applications more efficiently and effectively, and reducing the complexity of software integration.

Table 21: List of APIs

Standard Name	Full name	Description	Organisation	Open access	Year	Web Link
CEN WI 442032		Common Data Environments (CDE) for BIM projects –Open data exchange between platforms of different vendors via an open CDE API	CEN	/	/	(Henttinen, 2020)
DIN SPEC 91391	Common Data Environments (CDE) for BIM projects - Function sets and open data exchange between platforms of different vendors - Part 1: Components and function sets of a CDE; with digital attachment	Specify a concept for an open protocol for the exchange of data between two platforms without losses	The German Institute for Standardization (DIN)	No	2019	website
GS1 service	Global Standards 1	GS1 standards create a common foundation for business by uniquely identifying, accurately capturing and automatically sharing vital information about products, locations, assets and more. Businesses can also combine different GS1 standards to streamline business processes such as traceability.	The Global Standards One (GS1) organization	Yes	2005	website
oBIX	Open Building Information Exchange	open Building Information Exchange enables the mechanical and electrical control systems in buildings to communicate with enterprise applications, and to provide a platform for developing new classes of applications that integrate control systems with other enterprise functions.	OASIS Open	Yes	2006	website
OGC - FEATURES API		Features is a multi-part standard that offers the capability to create, modify, and query spatial data on the Web	OGC	Yes	2022	website

Standard Name	Full name	Description	Organisation	Open access	Year	Web Link
OGC 3D GEOVOLUMES API		Used for access and transfer of 3D geospatial content over the internet	OGC	Yes	2020	website
OGC API - PROCESSING		allows for processing tools to be called and combined from many sources and applied to data in other OGC API resources	OGC	Yes	2021	website
OGC API - RECORDS		offers the capability to create, modify, and query metadata on the Web	OGC	Yes	2021	website
OGC DISCRETE GLOBAL GRID SYSTEMS API		An API for accessing data organised according to a Discrete Global Grid System (DGGS).	OGC	Yes	2021	website
OGC MAPS API		Maps draft specification describes an API that can serve spatially referenced and dynamically rendered electronic maps	OGC	Yes	2022	website
OGC TILES - API		The OGC API — Tiles standard defines building blocks for creating Web APIs that support the retrieval of geospatial information as tiles.	OGC	Yes	2022	website
Uniclass API		The Uniclass API allows to access Uniclass in your own platform, making it even easier to apply to your day to day processes.	UK-based Royal Institute of British Architects (RIBA)	Yes	2015	website

Organizations such as OGC have created multiple APIs (see Table 21) to facilitate the interoperability and accessibility of building data on the web. For instance, the OGC-FEATURES API enables spatial data manipulation on the web, the OGC 3D-GEOVOLUMES API is used for transmitting 3D geospatial content online, OGC API-PROCESSING allows the calling and combining of processing tools from various sources, OGC API-RECORDS interacts with metadata online, OGC MAPS API interacts with electronic maps, and OGC TILES-API supports the retrieval of geospatial information as tiles. CEN WI 442032 API has been proposed by the CEN to facilitate an open data exchange between platforms of different vendors. The Open CDE workgroup has developed the BCF-API within buildingSMART International to enable interoperability with the BCF standard. The Global Standards One (GS1) organization has created the GS1 service to identify, capture, and share information about products, locations, and building assets. Additionally, the Uniclass API allows access to Uniclass[21] from other platforms, while the Open Building Information Exchange (oBIX) enables mechanical and electrical control systems in buildings to communicate with enterprise applications.

5.3 Summary

Given the reliance of the ACCORD automatic compliance checking process on the existing regulations and standards, it is crucial to outline the applicable standards within the scope of D2.1. This section aims to identify the standards that are relevant to ACCORD objectives. It explores and categorizes the standards related to building compliance checking. The findings of this section will contribute to various tasks within the project, such as T2.2 (Compliance Ontology) which focuses on creating the AEC3PO ontology. T2.2 can utilize the data models, terminologies, and taxonomies outlined in this section, adapting, or aligning them to enhance the AEC3PO ontology. The output of this section will also benefit T2.5 (Rule Formalisation Tool), which can leverage the recommended rule languages to formalize regulations based on the selected rule language. Additionally, the APIs presented in this section can be of interest to T4.3 (ACCORD Microservices, APIs) and T4.4 (Consortium Microservices, APIs), as they can support the development of the solution.

6. Reasoners and Rule Engines

Many studies (Doukari et al., 2022; Ismail et al., 2017; Lee et al., 2019; Li and Schultz, 2021; Pauwels et al., 2011; Zhang, 2018; Zhang and El-Gohary, 2016) have showed how reasoning is mandatory to the ACC systems. Authors in (Doukari et al., 2022) have discussed the reasoner solutions that use logic, such as Deontic Logic, Fuzzy Logic, or Answer Set Programming, to represent and/or execute the rules. The ACC system proposed in (Zhang and El-Gohary, 2016) relies on logic-based reasoning solutions to automatically reason about the logic rules and facts and generate compliance checking reports. The Ontological approach proposed in (Ismail et al., 2017) can use a reasoning functionality to match the model with the construction regulations. Beside those systems, authors in (Hu et al., 2021) have proposed a design support system using automatic rule checking to identify the compliance of rules and adopting case-based reasoning. This system purpose is to provide recommendations via ontology and semantics. Moreover, the study made in (Zhang, 2018) have shown that automated reasoning can also be used in Semantic Natural Language Processing-based Automated Compliance Checking (SNACC) system. SNACC allows the checking of quantitative

[21] https://www.thenbs.com/our-tools/uniclass

requirements from building codes. To automatically checks the logic facts with the logic rules and generates compliance checking reports.

Authors in (Pauwels et al., 2011) presented the two types of reasoning: the forward-chaining reasoners that starts from the available facts and uses the available inference rules to infer new facts. And the backward-chaining reasoner that starts from a set of hypotheses and the assertion that is to be proven.

We discus in this section the reasoning services that could be useful to building permitting automatic compliance checking. Then we present in a list of reasoners that could be used to validate the data, to apply inference rules, or to answer to conjunctive queries over a semantic data.

6.1 Reasoning Services

The reasoning is the main feature of semantic web technology. The difference between reasoning and querying is that querying allows us only to retrieve explicit facts whoever the reasoning process over a set of data and knowledge enables the inference of new RDF statements.

The most important reasoning services are:

- Classification: classifies a given concept in a concept hierarchy, for example: OuterDoor ⊑ Door.
- Concept satisfiability: verifies if a concept description does not necessarily designate the empty concept.
- Logical implication: checks whether a given relationship is a logical consequence of the schema description.
- Data validation: check if the data follow the defined shape, for example every person must have one and only one name, the process will not validate the instances with none or more than one name.

6.2 Reasoner's list

A semantic reasoner is a software system that is designed to make inferences and deductions based on a set of ontological axioms and rules. There are several well-known semantic reasoners that are commonly used. Table 22 contains the most known 47 reasoners with their supported rule language, DL or OWL 2 profile, the supported reasoning task (inference, data validation, and/or Query answering), and the publication/release year.

Table 22: List of reasoners

Reasoner	Rule language	Supported DL / OWL 2 profile	Inference	Validation	Query answering	year	Ref
Algernon	SWRL	OWL 2 EL	X			2006	website

Reasoner	Rule language	Supported DL / OWL 2 profile	Inference	Validation	Query answering	year	Ref
							(O'Connor et al., 2005)
BaseVISor	Triplet RDF and OWL facts with XML syntax	OWL 2 DL	X		X	2006	website (Matheus et al., 2006)
BUNDLE	OWL queries	Probabilistic description logic P-SHIQ(D)	X			2013	(Riguzzi et al., 2013)
CEL	Rules expressed in DL	EL++	X			2006	(Baader et al., 2006; Mendez and Suntisrivaraporn, 2009)
Chainsaw	OWL queries	OWL 2 DL	X			2012	(Tsarkov and Palmisano, 2012)
Closed World Machine (CWM)	N3Logic	Not a DL reasoner	X			Late 1990s	website
Clipper	SPARQL queries	Horn-SHIQ			X	2012	(Eiter et al., 2012)
DBOWL	SPARQL queries	OWL 1 DL	X		X	2008	(Roldan-Garcia and Aldana-Montes, 2008)
DistEL	Logic rules for classification	EL+	X			2013	(Mutharaju et al., 2013)
DRAOn	DL clauses	OWL 2 DL	X			2013	(Le Duc et al., 2013)
DReW	Datalog rules (support negation)	OWL 2 RL : LDL+ OWL 2 EL : SROEL(⊓ , ×)			X	2010	website (Xiao et al., 2010)
Drools	Drools rules	Not a DL reasoner	X			2005	website
ELepHant	DL clauses	EL+	X			2013	(Sertkaya, 2013)

Reasoner	Rule language	Supported DL / OWL 2 profile	Inference	Validation	Query answering	year	Ref
ELK	DL clauses	$EL_{\perp}^{+}$	X			2009	(Kazakov and Klinov, 2015)
Euler Yap Engine (EYE)	N3Logic	OWL 2 EL OWL 2 QL OWL 2 RL	X			2011	website
ELOG	log-linear description logics	EL++	X			2011	(Niepert et al., 2011)
FaCT++	DL clauses	SHOIQ	X			2005	(Tsarkov and Horrocks, 2006)
fuzzyDL	IF-THEN rules (DL)	fuzzy	X			2008	(Bobillo and Straccia, 2008)
HermiT	SPARQL queries SWRL	OWL 2 DL	X			2006	(Glimm et al., 2014)
HydrOWL	datalog rules	OWL 2 RL: SHIQ OWL 2 DL: ELHI			X	2014	(Stoilos, 2014)
Jcel	DL rules	EL+	X			2012	(Mendez, 2012)
Jess	Jess rules	Not a DL reasoner	X			2005	(O'Connor et al., 2005)
JFact	DL clauses	OWL 2 DL	X			2011	website
KARMA	Prolog	ELHO			X	2011	website
Konclude	DL-safe rules	SROIQV	X			2014	(Steigmiller et al., 2014)
LiFR	First order clauses	fuzzy	X			2014	(Tsatsou et al., 2014)
Living Semantic Platform	SPARQL queries	OWL 2 QL			X	2015	(Chen and Lambertz, 2015)

Reasoner	Rule language	Supported DL / OWL 2 profile	Inference	Validation	Query answering	year	Ref
Mastro	Datalog rules	DL-Lite	X		X	2010	(De Giacomo et al., 2012)
Mini-ME	Not a rule-based reasoner	ALCHIR+ ALN	X			2012	(Ruta et al., 2012)
ontop	SPARQL end-point (does not support SWRL yet)	OWL 2			X	2013	website
OwlOntDB	Datalog rules	OWL 2 EL: EL++ OWL 2 QL: DL-Lite OWL 2 RL: pD*	X			2013	(Faruqui and MacCaull, 2013)
OpenRulesEngine	OpenRules RuleML	Not a DL reasoner	X			2014	website
PAGOdA	Datalog rules	ALC	X		X	2014	(Zhou et al., 2015)
Pellet	SWRL	OWL 2 DL	X		X	2003	website
pySHACL	SHACL	Not a DL reasoner		X		2018	website
Racer	SWRL	ALC	X		X	2001	(Haarslev et al., 2012)
RDFox	Datalog SWRL	OWL 2	X			2015	website
RuQAR	SWRL	OWL 2 RL	X			2014	(Bak et al., 2014)
SHaclEX	SHACL ShEx	Not a DL reasoner		X		2016	website
Snorocket	DL clauses	ELK	X			2012	(Kazakov et al., 2012)
TORNADO	SWRL	OWL 2 EL	X			2016	website

Reasoner	Rule language	Supported DL / OWL 2 profile	Inference	Validation	Query answering	year	Ref
TReasoner	DL clauses	SHOIQ(D)	X			2013	(Grigorev and Ivashko, 2013)
TRILL	Prolog	SHOIN(D) ALC	X			2013	website
TRILLP	Prolog	SHOIQ	X			2015	(Zese et al., 2015)
TrOWL	DL clauses	OWL 2 DL	X		X	2010	(Thomas et al., 2010)
VIDRE	RuleML	/	X			2006	(Nagl et al., 2006)
WSClassifier (WSReasoner)	DL clauses	ALCHOI	X			2012	(Song et al., 2012)

6.3 Summary

We have presented in this section the reasoning task of ACC systems and the different engines used to perform the reasoning tasks, such as consistency checking, instance retrieval, and classification with ontologies expressed in OWL, which is a standard for representing ontologies in the Semantic Web. OWL is based on DL, and it provides a rich set of constructs for modelling complex concepts and relationships.

Some of the presented reasoners have specific strengths and weaknesses, so the choice of a reasoner depends on the particular requirements of the task and the size and complexity of the ontology being used. Some of the reasoners are specifically designed to perform one particular task such as **Clipper, DReW, HydrOWL, KARMA, Living Semantic Platform, and ontop** that was proposed for query answering so they can perform well even with a large set of data and a very expressive ontology. The expressivity of the ontology depends on the supported DL and the covered OWL profile. In general, DLs can be categorized into two main groups: expressive and lightweight.

Expressive DLs, such as OWL 2, provide a rich language for representing complex ontologies and allow for reasoning over multiple axioms. These DLs are more suitable for modelling large, complex domains but can come at the cost of decreased reasoning performance. Among the reasoners that support such expressive ontologies we find **BaseVISor, Chainsaw, DRAOn, HermiT, JFact, Living Semantic Platform, ontop, Pellet, RDFox, RuQAR,** and **TrOWL.**

Lightweight DLs, such as EL and DL-Lite, provide a simpler language and improved reasoning performance but limit the expressiveness of the ontology. These DLs are more suitable for modelling smaller, less complex domains. The reasoners that support this DL category, we have **Algernon, CEL, DistEL, ELepHant, ELK, ELOG, Jcel, Mastro,** and **TORNADO.**

In between these two groups, there are intermediate DLs such as SHOIN and SHOIQ which provide a good balance of expressiveness and reasoning performance. The reasoners supporting this DLs are **FaCT++, TReasoner, TRILL,** and **TRILLP.**

It is worth noting that some reasoners can cover more than one DL like **DReW** that supports OWL 2 RL with LDL+ fragment and OWL 2 EL with SROEL(⊓, ×) fragment. **Euler Yap Engine (EYE)** supports OWL 2 EL, OWL 2 QL, and OWL 2 RL. **HydrOWL** supports OWL 2 RL with SHIQ fragment and OWL 2 DL with ELHI fragment. **Mini-ME** with ALCHIR+ and ALN. **OwlOntDB supports** OWL 2 EL based on EL++, OWL 2 QL based on DL-Lite family, and OWL 2 RL inspired by pD∗.

BUNDLE reasoner uses a particular type of DL which is the probabilistic description logic P-SHIQ(D) that was proposed to handle uncertainty in OWL ontologies. With this particular DL, the axioms can be annotated with a set of variables and a probability between 0 and 1.

There is also Fuzzy Logic that is supported by some reasoners like **fuzzyDL** and **LiFR**. Fuzzy Logic is particularly useful for dealing with data that is not precise or can only be approximated but with less expressivity than OWL with P-SHIQ(D) logic.

Some reasoners are not based on a specific description logic but instead they use a different logic-based approach. For example: **Closed World Machine (CWM), Drools, Jess, OpenRulesEngine, pySHACL,** and **SHaclEX.** The last two reasoners are specifically proposed to validate the data model according to a defined data shape.

Unlike the other reasoners, **Mini-ME** is not a rule-based reasoner, but rather a DL reasoner. It was developed to perform semantic matchmaking for resource/service discovery in mobile and IoT devices, it is the first reasoner for mobile devices.

We have presented in this section a list of reasoners that are typically used for applications such as knowledge representation and reasoning, semantic web, and natural language processing . The choice of a specific reasoner depends on several factors, including the requirements of the application, the complexity of the ontology, and the performance requirements.

The reasoners and engines featured in this section cater to both key tasks in ACC, inference, and validation. They can be utilized to achieve the objective of ACC tasks, which involve validating building data and making inferences based on existing knowledge.

To facilitate the execution of rule reasoning or validation, we have provided a compilation and description of various reasoners and rule engines in this section. This information will be particularly valuable for WP4. It will aid in selecting the most appropriate reasoners and rule engines that align with the specific requirements of reasoning or validation tasks.

7. Conclusions

This deliverable has documented the existing standards, ontologies, query languages, rule languages, and reasoners that are relevant to automatic compliance checking in the AEC domain. The analysis and evaluation presented in this study provides valuable insights for other tasks in the project, such as the development of the AEC3PO ontology (T2.2) and the design and implementation of the Rule Formalisation Tool (T2.5), which can benefit from the recommendations and findings presented here.

The deliverable has achieved the goals of T2.1 by reviewing the existing ontologies, standards, and data models in the AEC domain and how they can be repurposed for automatic compliance checking and met the following objectives:

- Evaluation and analysis of AEC domain-related ontologies and how they can be employed for AEC3PO (Section 2).
- Recommendation of query languages according to their functions that depends on the type of data being queried (Section 3).
- Comparing the rule languages for better understanding of which rule languages are the most effective in terms of expressivity for representing regulations (Section 4).
- Reviewing the standards that are relevant to different areas of the ACCORD project, as demonstrated in Section 55.
- Presenting a comparative analysis of the available reasoners based on the supported DL and reasoning task (Section 6).
- Overall, this deliverable provides a comprehensive analysis and evaluation of the relevant elements in the AEC domain for automatic compliance checking and lays the foundation for the next steps in the ACCORD project.

Having achieved these objectives has generated valuable knowledge on standards, ontologies, query languages, rule languages, and reasoners that will prove beneficial for other tasks within the ACCORD project. This groundwork will facilitate the development of the AEC3PO ontology as well as the design and implementation of the Rule Formalisation Tool.

The accomplishments of each section have direct implications for specific tasks within the project, offering valuable contributions Section 2 will similarly contribute to T2.2 (Compliance Ontology) by providing ontologies. Query languages from Section 3 will prove beneficial for T5.1 (Automated BIM-based Building Permit and Environmental Compliance – Finland & Estonia Demo), T5.2 (Automated Checking for Land Use Permitting, Green Building Certification and Architectural Design Compliance of Industrialized Timber Housing), T5.3 (Automatic Checking of Structural Integrity of Steel Modular House Components – UK Demo), and T5.4 (Automated Checking of Compliance with Urban Regulations - Spain Demo). Rule languages discussed in Section 4 will support T2.3 (Machine-executable Regulations) and T2.5 (Design and Implementation of Rule Formalisation Tool). Section 5 will provide relevant inputs for T2.2 (Compliance Ontology), T2.5 (Rule Formalisation Tool), T4.3 (ACCORD Microservices, APIs), and T4.4 (Consortium Microservices, APIs) in terms of standards. Finally, Section 6 will provide valuable resources for T2.5 (Design and Implementation of Rule Formalisation Tool) in terms of reasoners and rule engines.

References

Abdelghani, M., 2021. Towards a Generalized Value Stream Mapping and Domain Ontology to Support the Enabling of Industry 4.0 in Construction (PhD Dissertation). https://doi.org/10.7939/r3-enqa-w715

Baader, F., Lutz, C., Suntisrivaraporn, B., 2006. CEL—a polynomial-time reasoner for life science ontologies, in: Automated Reasoning: Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006. Proceedings 3. Springer, pp. 287–291.

Bak, J., Nowak, M., Jedrzejek, C., 2014. RuQAR: Reasoning Framework for OWL 2 RL Ontologies, in: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (Eds.), The Semantic Web: ESWC 2014 Satellite Events, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 195–198. https://doi.org/10.1007/978-3-319-11955-7_17

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R., Kjærgaard, M.B., Srivastava, M., Whitehouse, K., 2016. Brick: Towards a Unified Metadata Schema For Buildings, in: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, BuildSys '16. Association for Computing Machinery, New York, NY, USA, pp. 41–50. https://doi.org/10.1145/2993422.2993577

Beach, T.H., Rezgui, Y., Li, H., Kasim, T., 2015. A rule-based semantic approach for automated regulatory compliance in the construction sector. Expert Systems with Applications 42, 5219–5231. https://doi.org/10.1016/j.eswa.2015.02.029

Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J., 2008. N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming 8, 249–269. https://doi.org/10.1017/S1471068407003213

Bobillo, F., Straccia, U., 2008. fuzzyDL: An Expressive Fuzzy Description Logic Reasoner, in: 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence). Presented at the 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pp. 923–930. https://doi.org/10.1109/FUZZY.2008.4630480

Bouzidi, K.R., Fies, B., Bourdeau, M., Faron-Zucker, C., Le-Thanh, N., 2013. An ontology for modelling and supporting the process of authoring technical assessments. arXiv preprint arXiv:1302.4726.

Bouzidi, K.R., Fies, B., Zucker, C.F., Zarli, A., Thanh, N.L., 2012. Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry. Future Internet 4, 830–851. https://doi.org/10.3390/fi4030830

buildingSmart Regulatory Room, 2017. Report on Open Standards for Regulations, Requirements and Recommendations Content.

Bus, N., Roxin, A., Picinbono, G., Fahad, M., 2018. Towards French Smart Building Code: Compliance Checking Based on Semantic Rules, in: Linked Data in Architecture and Construction (LDAC 2018), CEUR-WS Proceedings Vol. 2159. London, pp. 6–15.

Bushby, S.T., 2016. The Facility Information Model Standard- Guiding Facility Control into the Smart Grid World. NIST 58, 24–34.

Cao, J., Vakaj, E., Soman, R.K., Hall, D.M., 2022. Ontology-based manufacturability analysis automation for industrialized construction. Automation in Construction 139, 104277. https://doi.org/10.1016/j.autcon.2022.104277

Chen, W., Lambertz, J., 2015. Living semantic platform, in: Proceedings of the 11th International Conference on Semantic Systems, SEMANTICS '15. Association for Computing Machinery, New York, NY, USA, pp. 185–187. https://doi.org/10.1145/2814864.2814893

Clocksin, W.F., Mellish, C.S., 2003. Programming in Prolog. Springer Science & Business Media.

Daum, S., Borrmann, A., 2013. Checking spatio-semantic consistency of building information models by means of a query language, in: Construction Applications of Virtual Reality (CONVR 2013). London, UK.

De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R., Ruzzi, M., Savo, D.F., 2012. MASTRO: A reasoner for effective ontology-based data access, in: Proc. of the OWL Reasoner Evaluation Workshop (ORE 2012).

Di Martino, B., Mirarchi, C., Ciuffreda, S., Pavan, A., 2020. Analysis of Existing Open Standard Framework and Ontologies in the Construction Sector for the Development of Inference Engines, in: Barolli, L., Hussain, F.K., Ikeda, M. (Eds.), Complex, Intelligent, and Software Intensive Systems (CISIS 2019), Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 837–846. https://doi.org/10.1007/978-3-030-22354-0_77

Di Martino, B., Mirarchi, C., D'Abrunzo, F.F., Fiorillo, L., Iovine, N., Trebbi, C., Pavan, A., 2019. A Semantic and Rule Based Technique and Inference Engine for Discovering Real Estate Units in Building Information Models, in: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). Presented at the 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 81–88. https://doi.org/10.1109/AIKE.2019.00022

Dimyadi, J., Amor, R., 2013. Automated Building Code Compliance Checking – Where is it at? undefined.

Dimyadi, J., Pauwels, P., Amor, R., 2016. Modelling and accessing regulatory knowledge for computer-assisted compliance audit. Journal of Information Technology in Construction (ITcon) 21, 317–336.

Doukari, O., Greenwood, D., Rogage, K., Kassem, M., 2022. Object-centred automated compliance checking: a novel, bottom-up approach. Journal of Information Technology in Construction (ITcon) 27, 335–362. https://doi.org/10.36680/j.itcon.2022.017

Duque-Ramos, A., Fernández-Breis, J.T., Iniesta, M., Dumontier, M., Egaña Aranguren, M., Schulz, S., Aussenac-Gilles, N., Stevens, R., 2013. Evaluation of the OQuaRE framework for ontology quality. Expert Systems with Applications 40, 2696–2703. https://doi.org/10.1016/j.eswa.2012.11.004

E A de Mendonça, L Manzione, E Hjelseth, 2020. Converting Brazilian Accessibility Standard for BIM-Based Code Checking using RASE and SMC, in: Proceedings of the 37th International Conference of CIB W78. Sao Paulo, BR, pp. 291–307.

Eastman, C., Lee, Jae-min, Jeong, Y., Lee, Jin-kook, 2009. Automatic rule-based checking of building designs. Automation in Construction 18, 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Eilif Hjelseth, Beidi Li, 2021. Use of "Reasoner" for automated building code compliance assessment, in: Proc. of the Joint Conference CIB W78 - LDAC 2021. Luxembourg.

Eilif Hjelseth, Nick Nisbet, 2011. Capturing Normative Constraints by Use of the Semantic Mark-Up RASE Methodology. Presented at the CIB W78 2011 28th International Conference.

Eilif Hjelseth, Nick Nisbet, 2010. Exploring Semantic Based Model Checking, in: Proceedings of the CIB W78 2010.

Eiter, T., Ortiz, M., Simkus, M., Tran, T.-K., Xiao, G., 2012. Query Rewriting for Horn-SHIQ Plus Rules. Proceedings of the AAAI Conference on Artificial Intelligence 26, 726–733. https://doi.org/10.1609/aaai.v26i1.8219

El-Diraby, T.E., 2013. Domain Ontology for Construction Knowledge. Journal of Construction Engineering and Management 139, 768–784. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000646

El-Gohary, N., Asce, A., El-Diraby, T., 2010. Domain Ontology for Processes in Infrastructure and Construction. Journal of Construction Engineering and Management 136. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000178

Fahad, M., Bus, N., Andrieux, F., 2017. Towards Validation of IFC Models with IfcDoc and SWRL. Presented at the ICIW 2017, The Twelfth International Conference on Internet and Web Applications and Services, pp. 7–13.

Fahad, M., Bus, N., Andrieux, F., 2016. Towards Mapping Certification Rules over BIM, in: 33rd CIB W78 Conference. Brisbane, Australia.

Farias, T.M. de, Roxin, A., Nicolle, C., 2018. A rule-based methodology to extract building model views. Automation in Construction 92, 214–229. https://doi.org/10.1016/j.autcon.2018.03.035

Farias, T.M., Roxin, A., Nicolle, C., 2014. A Rule Based System for Semantical Enrichment of Building Information Exchange, in: RuleML 2014.
Faruqui, R.U., MacCaull, W., 2013. OwlOntDB: A Scalable Reasoning System for OWL 2 RL Ontologies with Large ABoxes, in: Weber, J., Perseil, I. (Eds.), Foundations of Health Information Engineering and Systems, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 105–123. https://doi.org/10.1007/978-3-642-39088-3_7
Fortineau, V., Paviot, T., Lamouri, S., 2019. Automated business rules and requirements to enrich product-centric information. Computers in Industry 104, 22–33. https://doi.org/10.1016/j.compind.2018.10.001
Future Insight Group, 2019. Introducing a Building Information Model (BIM)-based process for building permits in Estonia (No. Contract SRSS/C2019/024).
Gabriel Fierro, Karl Hammar, Joel Bender, Erik Paulson, Akshay Johar, Erik Wallin, Erik Wallin, 2022. On-demand webinar: A new standard is now available to use for harmonization between two smart building metadata standards. URL https://www.realestatecore.io/webinar-new-standard-brickrec/ (accessed 9.21.22).
Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z., 2014. HermiT: An OWL 2 Reasoner. J Autom Reasoning 53, 245–269. https://doi.org/10.1007/s10817-014-9305-1
Godager, B., 2018. Critical Review of the Integration of BIM to Semantic Web Technology, in: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Presented at the ISPRS TC IV Mid-term Symposium "3D Spatial Information Science – The Engine of Change" (Volume XLII-4) - 1–5 October 2018, Delft, The Netherlands, Copernicus GmbH, pp. 233–240. https://doi.org/10.5194/isprs-archives-XLII-4-233-2018
Gordon, T.F., 2010. An Overview of the Legal Knowledge Interchange Format, in: BIS (Workshops). Springer, pp. 240–242.
Grigorev, A.V., Ivashko, A.G., 2013. TReasoner: System Description., in: ORE. pp. 26–31.
Haarslev, V., Hidde, K., Möller, R., Wessel, M., 2012. The RacerPro knowledge representation and reasoning system. Semantic Web 3, 267–277. https://doi.org/10.3233/SW-2011-0032
Hayan Kim, Jaeyoung Shin, Jin-Kook Lee, 2018. How Can Computers Learn Building Design Rules?, in: Computational Studies on Cultural Variation and Heredity. https://doi.org/10.1007/978-981-10-8189-7_7
Henttinen, T., 2020. EUROPEAN BIM STANDARDS [WWW Document]. URL https://kirahub.org/wp-content/uploads/2020/02/12.00-European-BIM-standards-Tomi.pdf (accessed 3.16.23).
Hoekstra, R., Breuker, J., Di Bello, M., Boer, A., 2007. The LKIF Core Ontology of Basic Legal Concepts. LOAIT 321, 43–63.
Hoffmann, A., Shi, M., Wagner, A., Thiele, C.-D., Huyeng, T.-J., Rüppel, U., Sprenger, W., 2021. Evaluating SPARQL-based model checking: Potentials and limitations, in: ECPPM 2021 – EWork and EBusiness in Architecture, Engineering and Construction. CRC Press.
Hu, Z.-Z., Leng, S., Lin, J.-R., Li, S.-W., Xiao, Y.-Q., 2021. Knowledge Extraction and Discovery Based on BIM: A Critical Review and Future Directions. Arch Computat Methods Eng. https://doi.org/10.1007/s11831-021-09576-9
Ismail, A.S., Ali, K.N., Iahad, N.A., 2017. A Review on BIM-based automated code compliance checking system, in: 2017 International Conference on Research and Innovation in Information Systems (ICRIIS). Presented at the 2017 International Conference on Research and Innovation in Information Systems (ICRIIS), pp. 1–6. https://doi.org/10.1109/ICRIIS.2017.8002486
Jack, H., Wei Xi, X., 2019. Smart Grid/Building Semantic Integration for Interoperability, in: Workshop on Linked Building Data and Semantic Web Technologies (WLS2019). Smart Construction Informatics (SCI) Laboratory, Gainesville, Florida.
JAEYEOL SONG, JINSUNG KIM, JIN-KOOK LEE, 2019. Converting KBimCode Into an Executable Code for the Automated Design Rule Checking System, in: 24th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2019.
Jakob Beetz, Madhumitha Senthilvel, 2022. A Visual Programming Approach for Validating Linked Building Data, in: Lucian Constantin Ungureanu, Timo Hartmann (Eds.), EG-ICE 2020 Workshop on Intelligent Computing in Engineering.

Jiang, L., Shi, J., Wang, C., 2022. Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning. Advanced Engineering Informatics 51, 101449. https://doi.org/10.1016/j.aei.2021.101449

Kacfah Emani, C., Ferreira Da Silva, C., Fiès, B., Ghodous, P., Bourdeau, M., 2015. Automated Semantic Enrichment of Ontologies in the Construction Domain, in: 32nd CIB W78 Council for Research and Innovation in Building and Construction Conference. Eindhoven, Netherlands, pp. 159–164.

Kazakov, Y., Klinov, P., 2015. Advancing ELK: Not Only Performance Matters., in: Description Logics. pp. 233–248.

Kazakov, Y., Krötzsch, M., Simancik, F., 2012. ELK reasoner: architecture and evaluation., in: ORE. Citeseer.

Kifer, M., 2008. Rule Interchange Format: The Framework, in: Calvanese, D., Lausen, G. (Eds.), Web Reasoning and Rule Systems, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 1–11. https://doi.org/10.1007/978-3-540-88737-9_1

Kim, H., Lee, J.-K., 2016. Relational Logic Definition of Articles and Sentences in Korean Building Code for the Automated Building Permit System. Korean Journal of Computational Design and Engineering 21, 433–442. https://doi.org/10.7315/CDE.2016.433

Kim, H., Lee, J.-K., Shin, J., Choi, J., 2019. Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking. Journal of Computational Design and Engineering 6, 143–148. https://doi.org/10.1016/j.jcde.2018.08.002

Koh, J., Hong, D., Gupta, R., Whitehouse, K., Wang, H., Agarwal, Y., 2018. Plaster: an integration, benchmark, and development framework for metadata normalization methods, in: Proceedings of the 5th Conference on Systems for Built Environments, BuildSys '18. Association for Computing Machinery, New York, NY, USA, pp. 1–10. https://doi.org/10.1145/3276774.3276794

Körner, P., Leuschel, M., Barbosa, J., Costa, V.S., Dahl, V., Hermenegildo, M.V., Morales, J.F., Wielemaker, J., Diaz, D., Abreu, S., Ciatto, G., 2022. Fifty Years of Prolog and Beyond. Theory and Practice of Logic Programming 22, 776–858. https://doi.org/10.1017/S1471068422000102

Krijnen, T., Beetz, J., 2018. A SPARQL query engine for binary-formatted IFC building models. Automation in Construction 95, 46–63. https://doi.org/10.1016/j.autcon.2018.07.014

Krijnen, T.F., 2019. Efficient storage and retrieval of detailed building models: multi-disciplinary and long-term use of geometric and semantic construction information. Eindhoven University of Technology, Eindhoven, Netherlands.

Le Duc, C., Lamolle, M., Zimmermann, A., Curé, O., 2013. DRAOn: A Distributed Reasoner for Aligned Ontologies., in: ORE. pp. 81–86.

Lee, H., Lee, J.-K., Park, S., Kim, I., 2016. Translating building legislation into a computer-executable format for evaluating building permit requirements. Automation in Construction, The Special Issue of 32nd International Symposium on Automation and Robotics in Construction 71, 49–61. https://doi.org/10.1016/j.autcon.2016.04.008

Lee, J.K., 2011. Building environment rule and analysis (BERA) language and its application for evaluating building circulation and spatial program (PhD Dissertation). Georgia Institute of Technology.

Lee, J.-K., Eastman, C.M., Lee, Y.C., 2015. Implementation of a BIM Domain-specific Language for the Building Environment Rule and Analysis. J Intell Robot Syst 79, 507–522. https://doi.org/10.1007/s10846-014-0117-7

Lee, P.-C., Lo, T.-P., Tian, M.-Y., Long, D., 2019. An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning. KSCE J Civ Eng 23, 1952–1962. https://doi.org/10.1007/s12205-019-1750-2

Lefrançois, M., Kalaoja, J., Ghariani, T., Zimmermann, A., 2017. The SEAS Knowledge Model (No. Deliverable 2.2). ITEA2 12004 Smart Energy Aware Systems.

Li, B., Bhatt, M., Schultz, C., 2019. lambdaProlog (QS): functional spatial reasoning in higher order logic programming, in: 14th International Conference on Spatial Information Theory (COSIT 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Li, B., Schultz, C., 2021. A reasoning-based approach for checking performance-based codes in digital building permit processes.
Li, X., Yang, D., Yuan, J., Donkers, A., Liu, X., 2022. BIM-enabled semantic web for automated safety checks in subway construction. Automation in Construction 141, 104454. https://doi.org/10.1016/j.autcon.2022.104454
Li, Y., Zhao, Q., Liu, Y., Hei, X., Li, Z., 2021. Semiautomatic Generation of Code Ontology Using ifcOWL in Compliance Checking. Advances in Civil Engineering 2021, e8861625. https://doi.org/10.1155/2021/8861625
Lu, Y., Li, Q., Zhou, Z., Deng, Y., 2015. Ontology-based knowledge modeling for automated construction safety checking. Safety Science 79, 11–18. https://doi.org/10.1016/j.ssci.2015.05.008
Mads Holten Rasmussen, Jan Karlshøj, Maxime Lefrançois, Mathias Bonduel, Christian Anker Hviid, 2018. OPM: An ontology for describing properties that evolve over time, in: Linked Data in Architecture and Construction (LDAC 2018), CEUR-WS Proceedings Vol. 2159. London, pp. 24–33.
Mads Holten Rasmussen, Pieter Pauwels, Maxime Lefrançois, Georg Ferdinand Schneider, 2017. Building Topology Ontology (BOT) [WWW Document]. URL https://w3id.org/bot (accessed 4.6.23).
Matheus, C.J., Baclawski, K., Kokar, M.M., 2006. BaseVISor: A Triples-Based Inference Engine Outfitted to Process RuleML and R-Entailment Rules, in: 2006 Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'06). Presented at the 2006 Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'06), pp. 67–74. https://doi.org/10.1109/RULEML.2006.6
Mazairac, W., Beetz, J., 2013. BIMQL: An open query language for building information models. Advanced Engineering Informatics 27, 444–456. https://doi.org/10.1016/j.aei.2013.06.001
McGibbney, L.J., Kumar, B., 2013. A comparative study to determine a suitable representational data model for UK building regulations. Journal of Information Technology in Construction (ITcon) 18, 20–39.
McGlinn, K., Blake, D., O'Sullivan, D., 2019. GViz - An Interactive WebApp to Support GeoSPARQL over Integrated Building Information, in: Companion Proceedings of The 2019 World Wide Web Conference, WWW '19. Association for Computing Machinery, New York, NY, USA, pp. 904–912. https://doi.org/10.1145/3308560.3316536
Mendes de Farias, T., Roxin, A.-M., 2015. IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties, in: 32nd CIB W78 Conference on Information Technology in Construction. https://doi.org/10.48550/arXiv.1511.03897
Mendez, J., 2012. jcel: A Modular Rule-based Reasoner, in: 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), CEUR Workshop Proceedings.
Mendez, J., Suntisrivaraporn, B., 2009. Reintroducing CEL as an OWL 2 EL reasoner., in: Description Logics.
Morkunaite, L., Al-Naber, F.H., Petrova, E., Svidt, K., 2021. An Open Data Platform for Early-Stage Building Circularity Assessment, in: Linked Data in Architecture and Construction (CIB W78-LDAC 2021), Proceedings of the 37th International Conference of CIB W78. University of Ljubljana, Luxembourg, pp. 813–822.
Mutharaju, R., Hitzler, P., Mateti, P., 2013. DistEL: A Distributed EL+ Ontology Classifier. Presented at the SSWS@ISWC.
Nagl, C., Rosenberg, F., Dustdar, S., 2006. VIDRE - A distributed service-oriented business rule engine based on RuleML, in: IEEE Enterprise Distributed Object Computing Workshop (EDOC 2006). p. 44. https://doi.org/10.1109/EDOC.2006.67
Nawari, N., 2019. A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking Buildings, 17. https://doi.org/10.3390/buildings9040086
Nawari, N.O., 2018. Building Information Modeling: Automated Code Checking and Compliance Processes, 1st ed. CRC Press. https://doi.org/10.1201/9781351200998
Niepert, M., Noessner, J., Stuckenschmidt, H., 2011. Log-linear description logics, in: IJCAI. pp. 2153–2158.

NSAI TECHNICAL COMMITTEES NSAI/TC 68 BUILDING INFORMATION MODELLING (BIM) (ANNUAL REPORT No. 2021), 2021.

O'Connor, M., Knublauch, H., Tu, S., Grosof, B., Dean, M., Grosso, W., Musen, M., 2005. Supporting Rule System Interoperability on the Semantic Web with SWRL, in: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (Eds.), The Semantic Web – ISWC 2005, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 974–986. https://doi.org/10.1007/11574620_69

Pauwels, Pieter, de Farias, T.M., Zhang, C., Roxin, A., Beetz, J., De Roo, J., Nicolle, C., 2017a. A performance benchmark over semantic rule checking approaches in construction industry. Advanced Engineering Informatics 33, 68–88. https://doi.org/10.1016/j.aei.2017.05.001

Pauwels, Pieter, Krijnen, T., Terkaj, W., Beetz, J., 2017b. Enhancing the ifcOWL ontology with an alternative representation for geometric data. Automation in Construction 80, 77–94. https://doi.org/10.1016/j.autcon.2017.03.001

Pauwels, P., Roxin, A., 2017. SimpleBIM: From full ifcOWL graphs to simplified building graphs, in: Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM). pp. 11–18.

Pauwels, P., Terkaj, W., 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction 63, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., Van Campenhout, J., 2011. A semantic rule checking environment for building performance checking. Automation in Construction 20, 506–518. https://doi.org/10.1016/j.autcon.2010.11.017

Pauwels, P., Zhang, S., 2015. Semantic rule-checking for regulation compliance checking: an overview of strategies and approaches, in: Proceedings of the 32nd CIB W78 Conference on Information Technology in Construction. Presented at the 32nd International CIB W78 conference, pp. 619–628.

Pauwels, P., Zhang, S., Lee, Y., 2017. Semantic web technologies in AEC industry: A literature overview. https://doi.org/10.1016/J.AUTCON.2016.10.003

Poveda-Villalón, M., Espinoza-Arias, P., Garijo, D., Corcho, O., 2020. Coming to terms with FAIR ontologies, in: Knowledge Engineering and Knowledge Management: 22nd International Conference, EKAW 2020, Bolzano, Italy, September 16–20, 2020, Proceedings 22. Springer, pp. 255–270.

Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C., 2014. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. IJSWIS 10, 7–34. https://doi.org/10.4018/ijswis.2014040102

Preidel, C., Borrmann, A., 2017. Refinement of the Visual Code Checking Language for an Automated Checking of Building Information Models Regarding Applicable Regulations, in: Computing in Civil Engineering 2017. Presented at the ASCE International Workshop on Computing in Civil Engineering 2017, American Society of Civil Engineers, Seattle, Washington, pp. 157–165. https://doi.org/10.1061/9780784480823.020

Preidel, C., Borrmann, A., 2016. Towards code compliance checking on the basis of a visual programming language. Journal of Information Technology in Construction (ITcon) 21, 402–421.

Preidel, C., Borrmann, A., 2015. Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling. ISARC Proceedings 1–8.

Preidel, C., Daum, S., Borrmann, A., 2017. Data retrieval from building information models based on visual programming. Vis. in Eng. 5, 18. https://doi.org/10.1186/s40327-017-0055-0

Pruvost, H., Zeidler, F., 2022. Application of Ontologies for BACS Design Automation, in: Linked Data in Architecture and Construction (LDAC 2022).

Ramanathan, G., Husmann, M., 2022. Semantic Description of Equipment and Its Controls in Building Automation Systems, in: Groth, P., Rula, A., Schneider, J., Tiddi, I., Simperl, E., Alexopoulos, P., Hoekstra, R., Alam, M., Dimou, A., Tamper, M. (Eds.), The Semantic Web: ESWC 2022 Satellite Events, Lecture Notes in Computer Science. Presented at the ESWC

2022, Springer International Publishing, Cham, pp. 307–310. https://doi.org/10.1007/978-3-031-11609-4_47
Rasmussen, M.H., Lefrançois, M., Schneider, G., Pauwels, P., 2021. BOT: The building topology ontology of the W3C linked building data group. Semantic Web. https://doi.org/10.3233/SW-200385
Rasmussen, M.H., Pauwels, P., Hviid, C.A., Karlshoj, J., 2017a. Proposing a central AEC ontology that allows for domain specific extensions, in: Proceedings of the Joint Conference on Computing in Construction (JC3). Presented at the Joint Conference on Computing in Construction, pp. 237–244. https://doi.org/10.24928/JC3-
Rasmussen, M.H., Pauwels, P., Lefrançois, M., Schneider, G.F., Hviid, C.A., Karlshøj, J., 2017b. Recent changes in the Building Topology Ontology. https://doi.org/10.13140/RG.2.2.32365.28647
Rasmussen, M.H., Pauwels, P., Poveda-Villalón, M., Tiwari, S., García-Castro, R., Bourreau, P., Fies, B., Sschlenger, J., Daniotti, B., Törma, S., Thomas, M., Böhms, M., Sziebig, G., Font, A., Loscos, E., Vicente-Legazpi, P., Imbrechts, P., 2022. Ontologies and building digital twins (SPHERE project whitepaper).
Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., 2013. BUNDLE: A Reasoner for Probabilistic Ontologies, in: Faber, W., Lembo, D. (Eds.), Web Reasoning and Rule Systems, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 183–197. https://doi.org/10.1007/978-3-642-39666-3_14
Robaldo, L., 2021. Towards compliance checking in reified I/O logic via SHACL, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, ICAIL '21. Association for Computing Machinery, New York, NY, USA, pp. 215–219. https://doi.org/10.1145/3462757.3466065
Roldan-Garcia, M. del M., Aldana-Montes, J.F., 2008. DBOWL: Towards a Scalable and Persistent OWL Reasoner, in: 2008 Third International Conference on Internet and Web Applications and Services. Presented at the 2008 Third International Conference on Internet and Web Applications and Services, pp. 174–179. https://doi.org/10.1109/ICIW.2008.122
Ruta, M., Scioscia, F., Di Sciascio, E., Gramegna, F., Loseto, G., 2012. Mini-ME: the Mini Matchmaking Engine., in: ORE.
Schwabe, K., Teizer, J., König, M., 2019. Applying rule-based model-checking to construction site layout planning tasks. Automation in Construction 97, 205–219. https://doi.org/10.1016/j.autcon.2018.10.012
Seppo Törmä, Yuan Zheng, 2022. Digital Construction Ontologies (DiCon) (Ontology).
Sertkaya, B., 2013. The ELepHant Reasoner System Description., in: ORE. pp. 87–93.
Shen, Y., Xu, M., Lin, Y., Cui, C., Shi, X., Liu, Y., 2022. Safety Risk Management of Prefabricated Building Construction Based on Ontology Technology in the BIM Environment. Buildings 12, 765. https://doi.org/10.3390/buildings12060765
Solihin, W., Dimyadi, J., Lee, Y., 2018. In Search of Open and Practical Language-Driven BIM-Based Automated Rule Checking Systems, in: CIB W78 2018: IT in Design, Construction, and Management. pp. 577–584. https://doi.org/10.1007/978-3-030-00220-6_69
Solihin, W., Eastman, C., 2016. A knowledge representation approach in BIM rule requirement analysis using the conceptual graph. Journal of Information Technology in Construction (ITcon) 21, 370–401.
Song, W., Spencer, B., Du, W., 2012. WSReasoner: A Prototype Hybrid Reasoner for ALCHOI Ontology Classification using a Weakening and Strengthening Approach., in: ORE. Citeseer.
Steigmiller, A., Liebig, T., Glimm, B., 2014. Konclude: System description. Journal of Web Semantics, Semantic Web Challenge 2013 27–28, 78–85. https://doi.org/10.1016/j.websem.2014.06.003
Stoilos, G., 2014. Hydrowl: A Hybrid Query Answering System for OWL 2 DL Ontologies, in: Kontchakov, R., Mugnier, M.-L. (Eds.), Web Reasoning and Rule Systems, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 230–238. https://doi.org/10.1007/978-3-319-11113-1_20
Stolk, S., McGlinn, K., 2020. Validation of IfcOWL datasets using SHACL, in: Poveda-Villalón, M., Roxin, A., McGlinn, K., Pauwels, P. (Eds.), Proceedings of the 8th Linked Data in Architecture

and Construction Workshop, CEUR Workshop Proceedings. Presented at the Linked Data in Architecture and Construction, CEUR, Dublin, Ireland, pp. 91–104.

Syed Abdullah, N., Sadiq, S., Indulska, M., 2012. A Compliance Management Ontology: Developing Shared Understanding through Models, in: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (Eds.), Advanced Information Systems Engineering, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 429–444. https://doi.org/10.1007/978-3-642-31095-9_28

Terkaj, W., Pauwels, P., 2017. A Method to Generate a Modular ifcOWL Ontology, in: JOWO.

The EXPRESS Definition Language for IFC Development [WWW Document], 2019. URL https://standards.buildingsmart.org/documents/Implementation/The_EXPRESS_Definition_Language_for_IFC_Development.pdf (accessed 3.16.23).

Thomas, E., Pan, J.Z., Ren, Y., 2010. TrOWL: Tractable OWL 2 Reasoning Infrastructure, in: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (Eds.), The Semantic Web: Research and Applications, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 431–435. https://doi.org/10.1007/978-3-642-13489-0_38

Tsarkov, D., Horrocks, I., 2006. FaCT++ description logic reasoner: System description, in: Automated Reasoning: Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006. Proceedings 3. Springer, pp. 292–297.

Tsarkov, D., Palmisano, I., 2012. Chainsaw: a Metareasoner for Large Ontologies., in: ORE.

Tsatsou, D., Dasiopoulou, S., Kompatsiaris, I., Mezaris, V., 2014. LiFR: A Lightweight Fuzzy DL Reasoner, in: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (Eds.), The Semantic Web: ESWC 2014 Satellite Events, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 263–267. https://doi.org/10.1007/978-3-319-11955-7_32

Xiao, G., Heymans, S., Eiter, T., 2010. DReW: a Reasoner for Datalog-rewritable Description Logics and DL-Programs.

Xue, X., Zhang, J., 2022. Regulatory information transformation ruleset expansion to support automated building code compliance checking. Automation in Construction 138, 104230. https://doi.org/10.1016/j.autcon.2022.104230

Yurchyshyna, A., Faron-Zucker, C., Le Thanh, N., Zarli, A., 2010. Knowledge capitalisation and organisation for conformance checking model in construction. International Journal of Knowledge Engineering and Soft Data Paradigms 2, 15–32.

Yurchyshyna, A., Faron-Zucker, C., Le Thanh, N., Zarli, A., 2008. Towards an Ontology-enabled Approach for Modeling the Process of Conformity Checking in Construction., in: CAiSE Forum. Citeseer, pp. 21–24.

Zadeh, P.A., Wei, L., Dee, A., Pottinger, R., Staub-French, S., 2019. BIM-CITYGML data integration for modern urban challenges. J. Inf. Technol. Constr. 24, 318–340.

Zentgraf, S., Hagedorn, P., König, M., 2022. Multi-requirements ontology engineering for automated processing of document-based building codes to linked building data properties, in: IOP Conference Series: Earth and Environmental Science. IOP Publishing, Melbourne, Australia, p. 092007. https://doi.org/10.1088/1755-1315/1101/9/092007

Zese, R., Bellodi, E., Riguzzi, F., Lamma, E., 2015. Tableau Reasoners for Probabilistic Ontologies Exploiting Logic Programming Techniques., in: DC@ AI* IA. pp. 1–6.

Zhang, C., Beetz, J., 2016. Querying linked building data using SPARQL with functional extensions. Presented at the eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 11th European Conference on Product and Process Modelling, ECPPM 2016, pp. 27–34.

Zhang, C., Beetz, J., de Vries, B., 2018. BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. Semantic Web 9, 829–855. https://doi.org/10.3233/SW-180297

Zhang, J., 2018. A logic-based representation and tree-based visualization method for building regulatory requirements. Visualization in Engineering 5. https://doi.org/10.1186/s40327-017-0043-4

Zhang, J., El-Gohary, N., 2016. A Prototype System for Fully Automated Code Checking. Presented at the 16th International Conference on Computing in Civil and Building Engineering (ICCCBE 2016).

Zhao, Y., Yang, Q., 2021. Development of an ontology-based semantic building post-occupancy evaluation framework. Int. J. Metrol. Qual. Eng. 12, 19. https://doi.org/10.1051/ijmqe/2021019

Zheng, Z., Zhou, Y.-C., Lu, X.-Z., Lin, J.-R., 2022. Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. Automation in Construction 142, 104524. https://doi.org/10.1016/j.autcon.2022.104524

Zhong, B., Gan, C., Luo, H., Xing, X., 2018. Ontology-based framework for building environmental monitoring and compliance checking under BIM environment. Building and Environment 141, 127–142. https://doi.org/10.1016/j.buildenv.2018.05.046

Zhong, B.T., Ding, L.Y., Luo, H.B., Zhou, Y., Hu, Y.Z., Hu, H.M., 2012. Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. Automation in Construction 28, 58–70. https://doi.org/10.1016/j.autcon.2012.06.006

Zhou, Y., Grau, B.C., Nenov, Y., Kaminski, M., Horrocks, I., 2015. PAGOdA: Pay-As-You-Go Ontology Query Answering Using a Datalog Reasoner. Journal of Artificial Intelligence Research 54, 309–367. https://doi.org/10.1613/jair.4757

Zhou, Y., She, J., Huang, Y., Li, L., Zhang, L., Zhang, J., 2022. A Design for Safety (DFS) Semantic Framework Development Based on Natural Language Processing (NLP) for Automated Compliance Checking Using BIM: The Case of China. Buildings 12, 780. https://doi.org/10.3390/buildings12060780