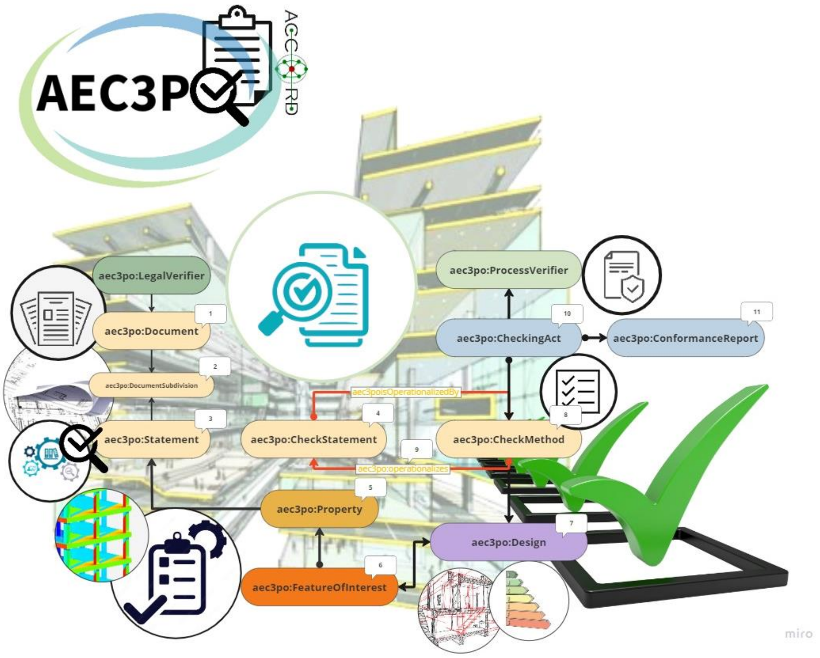
D2.2 BCO Ontology and Rules Format

22 December 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:	BCO Ontology and Rules Format
Type:	Report
Dissemination level:	Public
Work package:	WP2
Lead Beneficiary:	BCU
Deliverable leader:	Edlira Vakaj (BCU)
Contributing partners:	CU
Due date:	31.12.2023
Date:	22.12.2023
Status – version, date:	1.1 Final version – 22.12.2023
Authors:	Amna Dridi, Edlira Vakaj, Panagiotis Patlakas, Hansi Hettiarachchi (BCU)
	Thomas Beach, Jonathan Yeung (CU)
	Gonçal Costa (FUNITEC)
	Petr Hradil (VTT)
	He Tan (JU)
Reviewer 1:	Pasi Paasiala, Solibri
Reviewer 2:	Antoine Zimmermann, IMT

DOCUMENT HISTORY

Version	Date	% Complete	Comments	Main Authors (organisation)
0.1	09/08/2023	5	Initial Deliverable Structure and Table of Contents	Edlira Vakaj (BCU), Amna Dridi (BCU), Thomas Beach (CU)
0.2	23/10/2023	30	Section 3.1, 3.2 and 3.3	Amna Dridi (BCU)
0.3	25/10/2023	70	Section 2 and 4	Thomas Beach (CU)
0.4	30/10/2023	71	Section 2, NLP related content.	Hansi Hettiarachchi (BCU)
0.5	31.10.2023	89	Section 3.4, 3.5, 3.6, 5	Amna Dridi (BCU), Edlira Vakaj (BCU)
0.6	1.11.2023	94	Section 1, Summary, Conclusions, Internal Review	Edlira Vakaj (BCU)
0.7	2.11.2023	95	Revision and editing	Panagiotis Patlakas (BCU), Petr Hradil (VTT)
0.8	9.11.2023	96	Revision and editing	Gonçal Costa (FUNITEC), He Tan (JU)
0.9	1.12.2023	97	Internal Review	Pasi Paasiala (Solibri), Antoine Zimmermann (IMT)
1.0	15.12.2023	98	Proofreading and final internal review	Rita Lavikka (VTT), Jonathan Yeung (CU)
1.1 FINAL	22.12.2023	100	Submitted	Rita Lavikka (VTT)

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This deliverable contains original unpublished work except where clearly indicated otherwise.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 2.2 Building Compliance Ontology and Task 2.3 Machine Executable Regulations in the ACCORD project.

The aim of the ACCORD project is to digitalise 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 by adopting an approach where individual tools are treated as microservices, eliminating the requirement for costly centralised systems that are hard to establish and manage.

This deliverable reports on the methodology for *digitalising* and formalising regulations, the building compliance and permitting related ontology and the domain-specific rule language for expressing rules. The report builds on the outcomes of Deliverable 2.1 *Existing ontologies, standards, and data models in the building data domain relevant to compliance checking* and WP1 *Requirements for digitalising permitting and compliance processes*.

More specifically, this deliverable:

1. Provides a methodology for digitalising and formalising regulations and compliance-related documents.

2. Reports on the development of an ontology that captures key concepts, relationships and processes regarding compliance checking and permitting, namely the Architecture Engineering and Construction Compliance Checking Ontology (AEC3PO).

3. Reports on the development of a novel Domain Specific Language (DSL) for building compliance built upon AEC3PO.

4. Reports on the evaluation of the methodology and other artifacts documented in this deliverable, through their application in a series of test cases.

5. Reports on the creation of an ACCORD dictionary of terms from the ACCORD Demo Cases regulations to be mapped to BSDD.

The AEC3PO ontology feeds into the DSL rules, while feeding further into ACCORD WP4 and WP5. The vision for AEC3PO includes its continued expansion by the community, and it remains publicly accessible via the ACCORD GitHub workspace at: https://github.com/Accord-Project/aec3po.

Publishable summary

This deliverable presents the results of Task 2.2 Building Compliance Ontology and Task 2.3 Machine Executable Regulations in the ACCORD project.

The aim of the ACCORD project is to digitalise 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 by adopting an approach where individual tools are treated as microservices, eliminating the requirement for costly centralised systems that are hard to establish and manage.

This deliverable reports on the methodology for *digitalising* and formalising regulations, the building compliance and permitting related ontology and the domain-specific rule language for expressing rules. The report builds on the outcomes of Deliverable 2.1 *Existing ontologies, standards, and data models in the building data domain relevant to compliance checking* and WP1 *Requirements for digitalising permitting and compliance processes*.

More specifically, this deliverable:

6. Provides a methodology for digitalising and formalising regulations and compliance-related documents.
7. Reports on the development of an ontology that captures key concepts, relationships and processes regarding compliance checking and permitting, namely the Architecture Engineering and Construction Compliance Checking Ontology (AEC3PO).
8. Reports on the development of a novel Domain Specific Language (DSL) for building compliance built upon AEC3PO[1].
9. Reports on the evaluation of the methodology and other artifacts documented in this deliverable, through their application in a series of test cases.
10. Reports on the creation of an ACCORD dictionary of terms from the ACCORD Demo Cases regulations to be mapped to BSDD.

[1] w3id.org/lbd/aec3po/

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 The ACCORD Project

The objective of the ACCORD project is to provide a framework for the digitalisation of building permitting and compliance processes using Building Information Modelling (BIM), Geographic information system (GIS), and other data sources, with the end goal of improving productivity and quality of design and construction processes, supporting the design of climate-neutral buildings and advancing a sustainable built environment in line with the EU Green Deal and New European Bauhaus initiative.

ACCORD is based on the principle that these digitised processes must be human-centred, transparent, and cost-effective for the permit applicants and authorities and, above all, relevant to the industry within which they are to be employed.

To achieve this, ACCORD is developing a semantic framework for European digital building permitting processes, regulations, data, and tools. This framework will drive rule formalisation and integration of existing compliance tools as microservices. Solutions and tools are to be developed, providing consistency, interoperability and reliability with national regulatory frameworks, processes, and standards. It will enable the integration of technical solutions for automating compliance checking of buildings in their design, construction, and renovation/demolition lifecycle phases.

The Architecture, Engineering, and Construction (AEC) industry is subject to numerous regulations and standards that govern the design, construction, maintenance, operation, and demolition of buildings and infrastructure. These regulations often involve complex language and technical jargon, which can be difficult to understand and apply in practice. Semantisation, or the process of transforming natural language into machine-readable data with explicit meaning, can address this challenge by creating structured representations of regulations that can be exchanged and processed by computers. This can enable Automated Compliance Checking (ACC), facilitate communication between stakeholders, and improve the efficiency and effectiveness of regulatory enforcement.

The Semantic Web provides a framework for integrating and processing data from various sources, which can be used to automate compliance checking for buildings. Ontologies, Linked Data, Reasoners and rule-based systems are some of the Semantic Web technologies that can be applied to building compliance checking. Jointly, Artificial Intelligence (AI) methods such as Natural Language Processing (NLP) can be used to extract and analyse compliance requirements from natural language text in an automated way. However, there are several challenges in using Semantic Web technologies and NLP in building compliance checking, such as the need for accurate and complete data, the complexity of modelling compliance requirements, and the difficulty of interpreting natural language text.

One of the goals of WP2 (Semantisation of regulations and open format for machine-readable rules) is to develop an ontology that models building compliance requirements, including laws, regulations, processes, and documentation. The ontology requirements have been derived from the rule formalisation methodology and an extensive literature review conducted in D2.1. The methodology defines different pathways (manual, semi-automatic and automatic) for digitising and/or formalising rules, and subsequently populating the ontology with the extracted rules. The extracted rules can be encoded in Resource Description Framework (RDF) format to create a knowledge graph that can be used for ACC. This knowledge graph can be further enriched with rules extracted from the textual data, such as building codes, standards, and regulations. By leveraging the power of Machine Learning and NLP, the process of extracting rules from building regulation text can be automated, thereby reducing the time and effort required for compliance checking. A separate task is dedicated to the automatic extraction of rules from regulatory text. Despite the challenges of natural language, NLP and, more specifically, Large Language Models (LLMs) seem a very promising solution to automate rule extraction. On the other hand, the manual and semi-manual approach will be facilitated by the Graphical Rule Formalisation tool developed for a different task. The outputs of this WP will be used internally in the ACCORD compliance checking orchestration platform and shared with the community for further development and use.

1.2 Aims and Objectives

This deliverable reports the outcomes of Task 2.2 Building Compliance Ontology and Task 2.3 Machine Executable Regulations. The overall aim of this work is to provide methods and semantic models to digitalise and formalise building codes, regulations, and standards. These methods and models will be used to convert building codes, regulations, and standards into executable rule formats.

Objectives:

1. Develop a formal ontological model of building compliance requirements that is not specific to any particular region or legal system and ensure compatibility with established standard ontologies.
2. Analyse existing methodologies for producing machine-executable laws and regulations, drawing from academic literature and existing software systems.
3. Define the ACCORD methodology for making laws and regulations machine-executable, focusing on selecting and adapting existing approaches and integrating them with semantic definitions.
4. Design and apply a DSL language for expressing rules.
5. Evaluate the produced methodology, ontology, and DSL using ACCORD Demo Cases.

2. Regulation Digitalisation Methodology

This section documents the methodology developed within the ACCORD project to digitalise construction regulations. This methodological process will capture the entire process, starting from the original document, i.e., PDF, to a machine-readable document *(that software can read, parse, and understand the structure of)* to a fully machine-operable document (*that software can use to instigate a set of complex processes*) that can be used to drive digital building permitting processes within the ACCORD semantic framework.

In the remainder of the ACCORD project, the digitisation approach developed in this deliverable will be applied to 12 documents from the ACCORD demonstrators (shown in Table 1). However, the approach is generic and can be applied to any construction regulatory documents.

Regulation Name	Demonstration Case
Eurocode Structural Design	UK
Urban Planning Regulations	Spain
Population Information Systems	Finland
Accessibility	Finland

CO2 Emissions	Finland
Operational Safety	Finland
Land Use	Germany
Environmental Compliance	Germany
Regulations for Industrialised timber Housing	Germany
Accessibility	Estonia
Fire Safety	Estonia
Education - Health Protection	Estonia

Table 1 Selected Demonstration Regulations

The remainder of this Section will first present a summary of the state of the art in this field (Section 2.1). Section 2.2 will then describe the digitisation methodology in more detail. Finally, Section 2.3 will describe the verification that will be conducted within this task to verify the digitisation methodology.

2.1 State of the Art Summary

In the field of digitisation and automation of construction compliance checking generally, from analysis of the field conducted in D1.1 it can be seen three broad approaches have been taken: (1) side by side pairing of construction regulations and executable code (in various formats), (2) creation of mixed documents that fuse human readable and machine-readable elements and (3) use of automated translation approaches such as Natural Language Processing (NLP) to automate execution based on human readable documents.

This section will briefly describe the state of the art in each area, focusing on approaches for digitisation. A more complete review of the state of the art in terms of digital building permitting and automated compliance checking can be found in ACCORD Deliverable 1.1.

2.1.1 Pairing of Construction Regulations and Executable Code

The pairing of construction regulations and executable code primarily consists of manual and semi-automated approaches to convert human readable regulations into executable code in a variety of formats and languages.

Significant work in this field includes work by Tan et al. [1]. They proposed an approach to combine results from the hygrothermal performance simulation of a building envelope with building codes to support compliance checking. The approach relies on executable building codes being created manually in the form of decision tables derived from the targeted design regulations and their interdependencies.

New languages have also been developed for this task. For example, [2] presented a new domain-specific programming language, the Building Environment Rule and Analysis (BERA), to define, analyse and check rules. However, the use of this language required manual translation from regulatory documents. Melzner et al. [3] performed a case study of BIM-based ACC using decision tables, manually translated from regulatory texts, for early detection of fall hazards as part of the safety planning workflow. In the development of another custom language, Sydora and Stroulia [4] presented a domain-specific language for computationally representing building interior design rules only (non-regulation) and a method for evaluating rules in this language against a BIM model. Finally, [5] proposed the use of the Gherkin language for ACC, enabling them to leverage on technology from the domain of software development continuous integration.

Taking a different approach, Preidel and Borrmann introduced a semi-automated method for compliance checking using the Visual Code Checking Language (VCCL), which involves specifying, in a semi-automated way, regulations using the visual programming paradigm [6].

There have also been examples of manually constructing ontologies to perform ACC. Bus et al. [7] experimented with an approach based on semantic web technologies for compliance checking using the IfcOWL ontology. Their approach consisted of (a) homogenising the modelling style among different stakeholders of a project using a reference BIM Execution Plan, (b) creating regulatory terminology by enriching the IfcOWL vocabulary with explicit and inferred regulatory concepts, (c) simplifying the semantic representation of geometrical features by computing Industry Foundation Classes (IFC) object bounding boxes, (d) and generating machine processable regulatory requirements by semi-automatically converting natural language rules into SPARQL queries.

In other related work, Zhong et al. proposed a meta-model of construction quality inspection and evaluation concepts to overcome the large number of regulations in this area [8]. The meta-model is implemented as an ontology, which allows regulations to be expressed as a combination of OWL axioms and Semantic Web Rule Language (SWRL) rules. However, the manual specification of these regulations is required.

2.1.2 Mixed-Human and Machine-Readable Documents

Another approach commonly taken is the fusion of human-readable regulatory text and machine-executable meta-data, allowing a single document to represent human-readable and machine-executable regulations.

The most common representation used in this area is RASE. This was first used in 2011 to digitise extracts from the Norwegian accessibility standard, the Dubai building regulations and the US court design guidance document [9]. Subsequently, this was extended to facilitate the integration of regulation texts in BIM-based code checking tools [10].

RegBIM [10] then integrated their work as part of an end-to-end methodology for regulatory compliance, underpinned by using IFC as a data model. The methodology behind the software includes (a) the use of regulation experts to mark-up regulatory documents using RASE, (b) the use of BIM experts to map between the regulations and IFC data models, (c) the use of a rule engine, (later a semantic model) to perform the compliance checking, and (d) an innovative user interface to show the complex structure of compliance checking results to end users in an easily understood way.

Furthermore, Ciribini et al. presented an innovative use of model checking with a BIM-based e-procurement framework [11]. Their research methodology consisted of converting an existing set of tendering texts into computable rules using Solibri Office (following the RASE methodology) and tendering drawings into a BIM model using Revit.

2.1.3 Automated Translation

The final approach is the automated translation of regulatory documents into machine-executable code. This approach, due to the level of automation attained, negates the need to retain either a document that fuses machine-executable and human-readable regulations or to require the manual pairing of executable code with human-readable regulations.

One of the first pieces of work in this field was by Boukamp and Akinci, who conceptualised an approach to automatically extract inspection and quality control requirements from construction specifications, both specific and standardised [12]. Automating the interpretation of construction specifications enables consistent automation of subsequent inspection and/or defect detection tasks. The approach consists of two stages: first, identification of the components that would require inspection and associated tolerances and, second, evaluation of the deviations against captured as-built data (such as 3D point clouds). The authors point out that the process cannot, however, be fully automated due to the lack of required information available from the modelling standards and modelling tools, as well as the lack of support for contextual reasoning.

In further advancement of the field, Salama and El-Gohary proposed an approach to enrich the knowledge representation and reasoning of underlying compliance checking rules beyond commonly used if-then-else rules [13]. Also, in 2011, Zhang et al. implemented an automated object-oriented rule checker with a view to integrate safety planning in the design process for better project execution planning [14].

Using a similar approach, Zhang et al. developed algorithms for BIM-based automated safety checking [15]. The main contribution is a table-based safety rule translation algorithm. Their iterative rule-based checking methodology consists of 3 steps: (a) categorise the rule according to the identification of relevant objects and their geometrical attributes; (b) apply a safety checking algorithm on the objects using a rule engine, show checking results to inform the user; and (c) update the checking results following the user input and loop on the next object.

Zhang & El-Gohary [16] also used rule-based semantic natural language processing techniques to automate the extraction and the machine-process-able representation of regulatory requirements from textual regulatory documents. Their method was tested on several clauses from the International Building Code and evaluated by comparison with a manually generated reference. These authors were then able to identify sources of errors, that would allow to improve the accuracy of the automated checking capability.

Another study by Li et al. [17] also applied NLP coupled with spatial reasoning to automate utility compliance checking. In this work, the NLP algorithm translates the textual descriptions of spatial configurations into computer-processable spatial rules. In further developments in the NLP space, Zhang et al. [18] also presented an NLP-based methodology to semi-automate the generation of BIM extensions to support automated compliance checking. The methodology combined (a) part-of-speech pattern matching to extract regulatory concepts, (b) term-based matching and semantic-based matching to select relevant IFC concepts and machine-learning based classification to identify relationships between pairs of concepts. In related work, Roychoudhury et al. proposed an approach for semi-automated transformation of legal natural language (English) text to the Semantics of Business Vocabulary and Rules (SBVR) Model via authoring of Structured English (SE) rule [19]. The method relies on a domain dictionary and a clause-based open information extraction technique, a context-free grammar of SE and a framework for translating SE into conceptual regulatory models. SE allows the framework to benefit from the interactive input of domain experts.

In 2019, [20] and [21] defined a conceptual and theoretical framework to standardise the extraction of regulatory requirements from textual regulations for design review and propose a modular architecture to implement automated design reviews. This was then advanced to become the Generalised Adaptive Framework (GAF) [22]. GAF is a process for computerising regulatory compliance checking based on an object-based representation of building regulations. It enables the translation of regulations into efficient computable expressions. Using the GAF approach, Nawari et al. presented the development of a virtual permitting process for the state of Florida. Based on an analysis with local stakeholders, a virtual permitting framework is proposed using building information modelling [23]. This computable model, generated using the GAF approach, is then linked with a building information model.

Finally, Zheng et al. [24] use a mix of NLP and semantic alignment techniques to extract regulations from text documents and align the semantics found in the documents to those in an ontology that relates to IFC models. This then advances to an attempted automated generation of SPARQL queries based on this alignment.

2.2 ACCORD Digitisation Methodology

This section will describe the ACCORD digitisation methodology. Throughout this and the following sections, when discussing the methodology, the following digitisation terms are defined and contextualised:

- **Machine-readable:** Where software can read, parse, and understand the structure of regulation documents.
- **Machine-executable:** Where software can additionally execute actions based on the document.

Based on the literature reviewed in the previous section, the overarching approach of the ACCORD methodology will be a mixture of the approaches outlined in Section 2.1. This means it will consider:

1. A manual approach that will utilise a single document to represent both human-readable and machine-executable regulations.
2. An NLP-derived automated approach.
3. A hybrid approach that will be developed following the analysis of the NLP-based approach. It will combine elements of both previous approaches depending on the accuracy achieved by the NLP techniques.

The motivation for selecting a mixed methodology that will explore manual, automated and hybrid approaches is taken because the accuracy level of automated translation methods is not yet fully proven. Thus, a hybrid approach has the potential to provide some level of automated conversion where an acceptable level of accuracy can be achieved, coupled with manual refinement where acceptable accuracy cannot be achieved.

Given the hybrid approach being adopted, there will still be the necessity to provide a formal representation of the regulations. To this end, the methodology will utilise a hybrid human-readable/ machine-executable representation of the regulatory documents based on semantics. This will be described in Chapter 3.

The ACCORD methodology proposes five abstract steps, of which we will provide a manual, automated (NLP) and a hybrid approach. These steps are:

1. **Conversion to a machine-readable document:** This abstract step will convert documents that are not easily machine-readable (documents where the contents with their semantic meaning cannot easily be extracted i.e., PDF, Word etc.) into a machine-readable document to serve as input for the remaining abstract steps. It should be noted that some documents in the regulatory domain are already in an easily machine-readable format, in this case this step can be skipped.
2. **Identify clauses and formalise logical relationships:** This methodological step will seek to identify paragraphs and clauses that contain regulatory content. It will then identify and formalise how these paragraphs and clauses interact logically (e.g., providing AND/OR etc. relationships).
3. **Identify rules and formalise logical relationships:** This methodological step continues from the previous step. It will formalise the words or phrases within each previously identified clause/paragraph that indicates a regulatory decision. Based on this, the logical relationship between each identified word/phrase will be specified.
4. **Formalise Rules:** In many cases, each identified word/phrase from the previous step must be formalised into some form of logical decision. In many cases, this is a simple mapping of a term to a value (e.g., *width == 5).* However, in certain cases, due to the complex language used in regulatory documents, each word/phrase may well have more complex logical relationships involving several terms.
5. **Map words to execution context:** Finally, each term identified from the previous phrases must be mapped to an execution context, i.e., it must be explicitly specified how the data

associated with that term can be retrieved or calculated. This phase of the methodology will formalise these mappings.

A full description of this methodology is shown in Figure 1 and will be described in the subsections below.

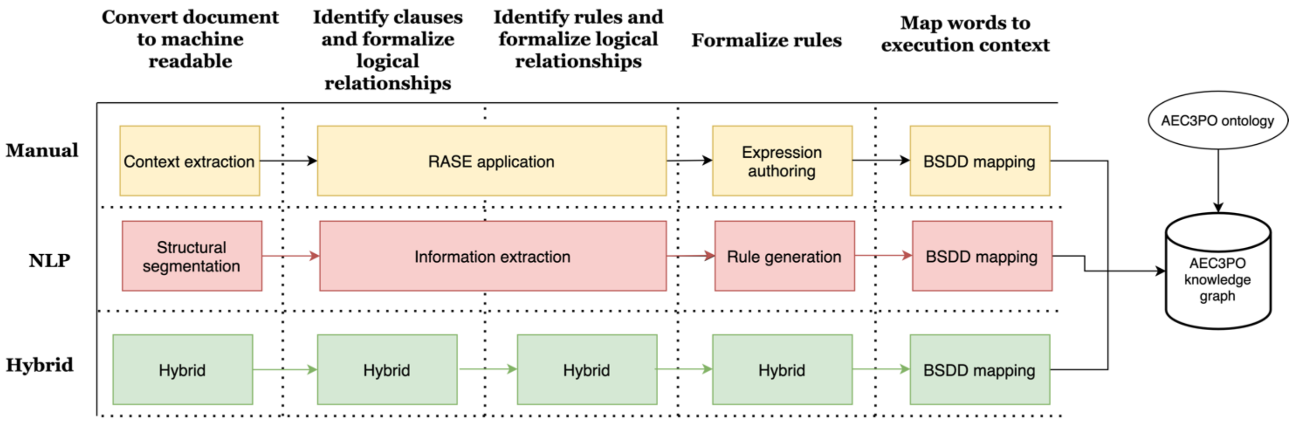


Figure 1 ACCORD digitisation methodology

2.2.1 Approach 1 (Manual)

The ACCORD manual methodology consists of the following steps that will each be described in more detail in Section 4

- **Content Extraction:** The regulation to be considered will be manually transformed into a spreadsheet-based format that can then be automatically parsed to produce initial input for the manual digitisation methodology.
- **RASE Application:** The identification of clauses, paragraphs and individual words and phrases, as well as the specification of their logical relationships, will be done using the RASE methodology. This is described in more detail in Section 4.3 RASE Application.
- **Expression Authoring:** This step will formalise the extraction of individual logical decisions. In most cases, this will be as simple logical operations; however, to deal with complex cases, a simple expression language will be proposed for use.
- **bSDD Mapping:** The mapping to the execution context will be performed using the BuildingSMART data dictionary; terms identified in the previous steps of the methodology will be mapped to terms within the bSDD. The definitions stored within this data dictionary will inform how the data associated with those terms can be retrieved/calculated.

2.2.2 Approach 2 (NLP)

The ACCORD automated methodology will be driven by NLP. It should be noted that the results of the NLP work are outside the scope of this deliverable and will be described in ACCORD Deliverable 2.3. However, a summary of the anticipated components is given here for completeness:

- **Structural Segmentation:** This component will separate a document into segments/blocks (e.g., sections, headings, paragraphs, sentences, etc.) based on its structure. These segments will be further processed to capture their interconnections and logical structures, which are required to extract the information described.
- **Information Extraction:** This component will extract information from different text segments/blocks. There will be three main targets under information extraction: (1) identifying

informative text blocks, (2) identifying entities described in informative text blocks, and (3) identifying relationships/connections between entities to convert the information described in natural language into a machine-readable format, which can facilitate the rule generation.

- **Rule Generation:** This component will map the information extracted from the text (e.g., content structure, entities, and relationships) with a rule format to generate the rules.
- **bSDD Mapping:** This element of the NLP methodology will not be automated and will be performed manually as per 2.2.1

2.2.3 Towards a Hybrid NLP/RASE Approach

There will also be a hybrid approach. This Hybrid approach consists of using the results of the NLP process to assist in some of the manual process activities when an automated process cannot be guaranteed. This is out of the scope of this deliverable, as it depends on the results of the NLP development and will be described in more detail in ACCORD Deliverable 2.3.

2.2.4 Introduction to AEC3PO Ontology

Architecture Engineering Construction Compliance Checking and Permitting Ontology - AEC3PO, integral to the compliance and permitting semantic framework within the ACCORD project, is a pivotal element of Work Package 2 (WP2), particularly in Task T2.2. Task T2.1 laid the foundation with a comprehensive literature review, and AEC3PO draws inspiration from the ontologies presented in T2.1 [25]. This ontology serves as a structured knowledge representation, capturing essential concepts, relationships, and rules pertaining to compliance checks and permitting stages within construction projects. Aligned with ACCORD's objectives, AEC3PO forms the basis for rule formalisation methodology (Task 2.3), Domain-Specific Rule Language, and a rule formalisation tool (Task 2.5). This facilitates seamless communication and collaboration among experts, stakeholders, and regulatory bodies in the Architecture, Engineering, and Construction (AEC) industry.

AEC3PO addresses various competency questions related to compliance and permitting, such as defining metadata for compliance checking documents, representing cross-referencing, identifying types of check statements, determining required data for checks, and transforming normative documents into implementable rules. For permitting, it answers questions about the stages of the permitting process, required evidence, associated standards, and necessary checks and verifications. These questions guide the development of AEC3PO, ensuring its relevance and applicability in real-world scenarios.

AEC3PO's design revolves around different modules such as the *Document*, CheckMethod, Feature of Interest modules, *etc*. Each module’s classes and properties are tailored to model distinct components and relationships in the construction domain. These modules provide a structured framework for representing compliance and permitting aspects, enhancing the ontology's versatility and effectiveness in addressing the complexities of the AEC industry. More details about AEC3PO will be presented in Section 3.

2.3 Verification

Within the context of WP2, both the digitisation methodology and the AEC3PO ontology must be verified. This will provide initial assurance that the methodology and ontology that have been specified are suitable for further use and eventual validation in the ACCORD demo projects (documented in WP5).

This verification will aim to (1) ensure that the digitisation methodology is able to capture the complexity of the regulatory documents that are likely to be encountered in the demonstration cases and (2) ensure that the representation of the results of the digitisation process as an instance of the AEC3PO ontology can correctly represent both human-readable and machine-executable aspects.

This verification will be performed in three steps:

1. The AEC3PO ontology has been evaluated based on an end-user evaluation (through a survey) and through the development of example instances of the ontology based on the demonstration cases.
2. The digitisation methodology will be verified by the manual digitisation process of two example regulations drawn from the demonstration cases.
3. This manual digitisation will also further verify the AEC3PO ontology by verifying that it can correctly represent the output of digitisation process. To support this, software tools will be developed to produce human-readable and machine-operable output from the instances of the AEC3PO ontology that are the results of the digitisation process.

This verification process will provide a solid base for the deployment and validation of the methodology as part of the ACCORD demonstration performed in WP5.

3. AEC3PO Ontology

This section presents a comprehensive exploration of the AEC3PO ontology, offering valuable insights into its development, structure, alignment, applicability, and evaluation. This section is divided into seven main subsections, each serving a distinct purpose. The first subsection (Section 3.1) initiates the discussion by conducting a thorough review of existing ontologies in the domain of compliance checking. This review establishes the foundational knowledge of prior work in the field, setting the stage for the development and evaluation of AEC3PO.

The second subsection (Section 3.2) illuminates the pivotal role of AEC3PO within the broader context of the ACCORD project, unveiling its intricate interconnections with various work packages and tasks. The third subsection (Section 3.3) explains the systematic approach employed to develop AEC3PO, emphasising the key phases and steps of ontology development. The fourth section (Section 3.4) provides an overview of the AEC3PO ontology, presenting its core structure, modules, classes, and properties. This section offers readers a high-level understanding of the organisation of the ontology and outlines the alignments between AEC3PO and relevant standards and ontologies, providing a comprehensive picture of the role of the methodology in shaping the ontology.

The fifth subsection (Section 3.5) showcases how the AEC3PO ontology is instantiated and applied in real-world scenarios. It discusses the applicability of the ontology by presenting cases from demo countries, demonstrating the practical use of AEC3PO in compliance checking tasks. The sixth subsection (Section 3.6) of this section focuses on evaluating AEC3PO through a variety of methodologies. These evaluations encompass both quantitative and qualitative analyses, assessing the performance, quality, and usability of the ontology. The results provide valuable insights into the effectiveness and reliability of AEC3PO.

Together, these subsections offer a holistic perspective on the AEC3PO ontology, from its inception and development to its real-world application and rigorous evaluation. This multifaceted approach ensures that AEC3PO is not only well-constructed but also highly effective in supporting compliance-checking processes within the AEC domain. Finally, the last subsection (Section 3.7) summarises the whole ontology section.

3.1 State of the Art Summary of Building Compliance Ontologies

The literature review on existing ontologies concerning compliance checking and digital permitting in the AEC domain highlights a range of research contributions that explore various dimensions of rule generation and compliance. These studies have significantly informed the development of the AEC3PO ontology. They also underscore certain limitations and gaps that have driven the need for proposing a more comprehensive and inclusive ontology in the context of building regulations.

This section will briefly describe the state of the art of relevant ontologies, focusing on compliance checking. The most notable aspects of the study carried out and reported in Deliverable 2.1 [25] are summarised in this section.

One of the first ontologies that have been proposed was the CQIEOntology – *Construction Quality Inspection and Evaluation Ontology* [8] – which streamlines construction quality compliance checking against regulations. This ontology facilitates inspection tasks by translating regulation provisions into inspection reminders. However, its focus is primarily on quality compliance, and it does not address the broader spectrum of building regulations.

A few years later, Zhong et al. [26] proposed the *Building Regulation Ontology* as part of their framework to support environmental monitoring and compliance checking under a building information modelling (BIM) environment. Their framework represents a network of four ontologies, namely the *Building Information Ontology*, the *semantic sensor network (SSN) ontology*, the *Building Regulation Ontology*, and the *Building Environmental Monitoring Ontology*. These ontologies integrate building information from BIM's, environmental information provided by sensors, and regulatory information based on building regulations and design requirements to ensure the building's environmental performance. While the framework showcases its effectiveness in a real distributed energy station project, it primarily centres around environmental aspects and lacks a comprehensive representation of building regulations beyond the environmental domain.

Similarly, the *code ontology* introduced in [27] serves as a key component in an automated code compliance-checking methodology. It comprises four primary ontologies: a *code ontology*, a *design model ontology,* a *merged ontology*, and a *code compliance checking ontology*. The *code ontology* is intended to encapsulate knowledge from building codes and establish a clear structure for information used in the compliance checking process. However, some limitations of this methodology are worth noting. For instance, *code ontology* primarily focus on specifying the types of information and their organisation but not the full scope of compliance rules. This could limit its adaptability to different regulatory systems and the automation of compliance checking for a broader range of AEC projects.

Within a comparable framework, Morkunaite et al. [28] have developed the *Building Circularity Assessment Ontology* (BCAO) that introduces a semantic modelling approach for construction quality inspection and evaluation against regulations, a valuable contribution to enhancing construction quality. However, it focuses primarily on quality checking and does not address the broader context of AEC compliance requirements. Within the same context, recently Fauth and Seiss [29] have designed the *Ontology for Building Permit Authorities* (OBPA) to provide a structural foundation for representing the organisational structure of building permit authorities, facilitating decision-making processes, and ensuring transparency and objectivity in the assignment of building permit applications. The ontology focuses on enabling the seamless exchange of information between building permit authorities, building officials, and applicants. It encompasses the representation of various entities, their relationships, and attributes within the building permit processing context. However, its scope is primarily limited to the assignment process within the building permit procedure.

In a more particular context, Li et al. [30] have proposed the *Railway Code Ontology* following a semi-automatic construction method based on ifcOWL. The ontology was developed by converting ifcOWL, which extends the semantic information of railway code. The extended ifcOWL is then converted to code ontology. The approach appears promising for specific domains like railway construction but may not be easily adapted to broader AEC compliance rules. The Safety Regulation Ontology (SRO) was developed within the same context to provide a semantic schema for subway construction safety checking [31], focusing on safety risk factor compliance. Although efficient, SRO remains focused on safety regulations, making it more domain-specific.

These research works collectively emphasise the importance of ontology-based compliance checking while underscoring the need for a more comprehensive ontology encompassing a wider range of building regulations, offering a unified foundation for rule generation and compliance checking across the AEC industry. AEC3PO ontology aims to address this need by providing a holistic and standardised solution.

3.2 The need for AEC3PO in ACCORD

The creation of a building compliance checking-specific ontology within the ACCORD framework serves a critical purpose in enabling interoperability and harmonising knowledge from diverse sources, including data extraction for building codes, Information Delivery Specification (IDS), and BIM and non-BIM reports, among others, within the ACCORD architecture. Tasks 2.2 and 2.3 present a significant challenge of correlating rule formalisation methodology that aims to semantise regulations and provide an open format for machine-readable and machine-executable rules. The *Architecture, Engineering and Construction Compliance Checking and Permitting Ontology (AEC3PO)* acts as the primary interface connecting knowledge-providing (i) building codes, regulations, and standards, (ii) compliance and permitting processes and documentation, and (iii) compliance and permitting actors.

Various methods, corresponding to the structure of the established classes within the ontologies reviewed in T2.1, and experts' inputs are considered. The Building Compliance Rule Language (BCRL), as part of Task 2.3, populates the ontology with instances of building compliance elements, such as the regulations documents, the statements that state the rules, the checking methods, etc. This process is effectively guided by the RASE (Requirement, Application, Selection, Exception) methodology, thoroughly described in Section 4, which provides a structured framework for capturing and categorising statements within these four key tags. Specifically, AEC3PO defines distinct classes and categories to represent statements, including *RequirementStatement*, *ApplicationStatement*, *SelectionStatement*, and *ExceptionStatement*. This approach ensures a meticulous and standardised representation of building elements and their associated rules, enhancing data quality and facilitating more effective rule generation and compliance checking processes.

All instances related to Building Regulations, conveying building compliance checking methods, acts and actors are obtained from the use cases of demo countries in WP5. In the next project stage, the AEC3PO will be used in (i) the implementation of the rule formalisation tool (WP2-T2.5) and APIs (WP4), (ii) the semantic mapping between the terms extracted by the NLP techniques developed in WP2-T2.4 and the ontological concepts, and (iii) the modelling of demo cases (WP5).

The conceptualisation phase of the AEC3PO places a strong emphasis on achieving a high level of accuracy. AEC3PO aims to enable reasoning capabilities while maintaining alignment with industry standards and existing ontologies. Consequently, efforts have been made to reuse well-known ontologies like Dublin Core Terms (DCT), Europe's Legislation Identifier (ELI), and more, enhancing the creation of a structured and interconnected knowledge graph. This allows professionals and machines to explore, query, and understand various aspects of the compliance and permitting processes more comprehensively.

Figure 2 provides a visual representation of the role of AEC3PO and its relationships with the knowledge application components of the ACCORD architecture. As AEC3PO serves as a semantic foundation for rule formalisation language in Task 2.3, it plays a vital role in API development (WP4), and it is integral to prototyping solutions for the demo cases (WP5). This interconnectedness underscores the significance of the ontology in enhancing the overall project's success and achieving seamless integration between various project components.

In this deliverable, we focus on the role of AEC3PO in the digitalisation methodology, namely in the development of the DSL.

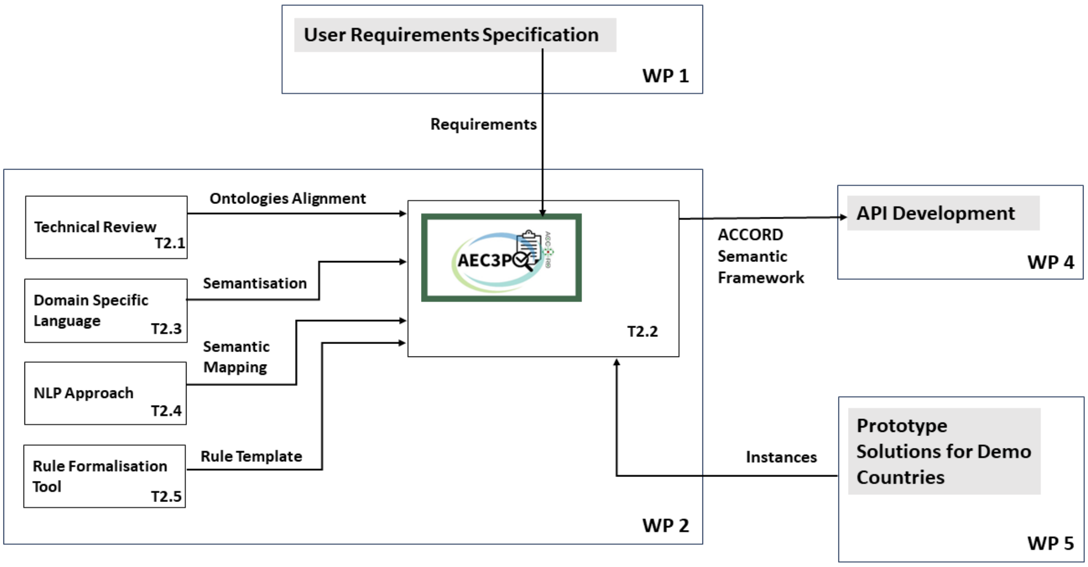


Figure 2 Role of AEC3PO in ACCORD Framework

3.3 AEC3PO Development Process

This section delves into the intricacies of the development process of the ontology, navigating through distinct phases that have sculpted its form and functionality. Emphasising pivotal aspects, such as data acquisition and ontology development methodology, this section explains the systematic approach employed to develop AEC3PO.

3.3.1 Overview of AEC3PO Development Process

In the ACCORD project, the development of AEC3PO ontology can be divided into four main phases, i.e. *(i) requirements analysis*, *(ii) conceptualisation*, *(iii) specification of concepts and relationships*, and *(iv) population, as shown in Figure 3*.

- **Requirements analysis:** The foundation of AEC3PO lies in its rigorous requirements analysis, a critical phase that shapes the purpose and scope of the ontology. These requirements have been meticulously drawn from multiple sources within the Horizon Europe project ACCORD, reflecting a comprehensive understanding of the needs and expectations of its stakeholders. The synthesis of these requirements emanates from three primary sources: (1) WP1 *User Requirements Specification*, which captures the essential insights and expectations of end-users and domain experts; (2) WP2 *T2.1 Existing Literature,* where an extensive review of relevant academic and industry literature provides valuable insights into best practices and existing ontologies in both building environment in general and building compliance checking in particular; and (3) expert interviews, which involve engaging with domain experts to extract valuable domain-specific knowledge for building permit and compliance process. This mixture of inputs ensures that AEC3PO is not merely a product of theoretical abstraction but a practical and valuable resource designed to address real-world challenges within the AEC domain. From the previous three sources, three main requirements have been derived that essentially concern modularisation:

    - Building codes, regulations, and standards.

    - Compliance and permitting processes and documentation.

    - Compliance and permitting actors.

2. **Conceptualisation:** The conceptualisation phase for AEC3PO involves the initial planning and high-level design of the ontology, focusing on the identification and definition of its key modules. In this phase, the goal is to establish a clear and comprehensive framework that will serve as the foundation for the development of the ontology. This phase involves four main steps:
    - ***Module identification:*** this first step is to identify the important modules that will be part of the AEC3PO ontology. These modules should align with the project's objectives and the specific requirements gathered from WP1, WP2, and expert interviews. In this step, five main modules have been identified to fulfil the requirements stated above: *Document*, *Feature of Interest*, *Design*, *Checking Method*, and *Checking Act*. These modules have been further modularised into other modules to generate a total of thirteen modules. More details on these models will be provided in Section 3.2.3. The decision to modularise the five modules of AEC3PO into further sub-modules rather than following a monolithic approach derives from several advantages associated with a modular design. Modularity enhances the maintainability, scalability, and reusability of the ontology. Breaking down AEC3PO into smaller, more manageable modules makes it easier to understand, modify, and extend specific functionalities without affecting the entire ontology. This approach promotes collaboration, as different teams or individuals can work on separate modules concurrently. Moreover, modularisation facilitates better organisation, making locating and addressing issues within specific components simpler. A modular structure provides flexibility and adaptability, allowing our proposed ontology to evolve efficiently as new requirements and use cases emerge. On the other hand, the decision to adopt a modularised approach for the five modules of AEC3PO is also driven by the goal of fostering the reuse and integration of existing ontologies. Modularisation enables the easy incorporation of external ontologies or modules that align with specific aspects of the AEC domain. This promotes interoperability and ensures AEC3PO can seamlessly leverage well-established ontologies or modules,

enhancing its overall comprehensiveness and utility. Additionally, a modular design allows for the targeted reuse of specific components, reducing redundancy and promoting a more efficient use of resources. This approach aligns with best practices in ontology engineering and contributes to the overall adaptability and extensibility of AEC3PO.

- ***Module definition:*** Each identified module should be defined in terms of its scope, purpose, and the types of information it will encompass. This step involves clarifying the boundaries and responsibilities of each module. The literature and domain experts have been involved in defining the thirteen modules and their related classes and properties.
- ***Module Interactions:*** Determine how the identified modules will interact with each other within the AEC3PO ontology. Identify the relationships, dependencies, and connections between all the modules to ensure a coherent and integrated ontology structure.
- ***Data Sources and Integration:*** Identify the sources of data and information that will feed into each module. This includes considering data from WP1, WP2, expert interviews, and external sources.

3. **Specification of Concepts and Relationships:** In the AEC3PO development, the specification of concepts and relationships is a crucial stage that involves the detailed definition and expansion of key concepts and relationships within the domain of Architecture, Engineering, and Construction (AEC). In this phase, the task aims to create a comprehensive and well-structured representation of the relevant knowledge, which can then be utilised for various purposes, including rule formalisation and automated compliance checking. This phase involves the following key activities and objectives:

   - *Concept Definition:* During this phase, the project team identifies and defines essential concepts related to the AEC domain in collaboration with AEC experts. These concepts can include building components, regulations, standards, processes, and any other elements relevant to the AEC industry.
   - *Relationships Between Concepts:* Besides defining individual concepts, the phase focuses on specifying the relationships between these concepts. This involves determining how concepts are connected or interrelated within the AEC domain. For example, it may establish relationships between building regulations and building components or checking methods and checking acts.
   - *Ontology Development:* The phase leads to the creation of an ontology or knowledge graph. An ontology formally represents the concepts and their relationships [32]. This ontology serves as a knowledge model for the AEC domain, providing a common vocabulary and semantic framework for understanding and processing information. The Agile and Continuous Integration for Modular Ontologies and Vocabularies (ACIMOV) ontology engineering methodology [33] has been adopted to develop AEC3PO. More details are available in Section 3.3.
   - *Knowledge Expansion:* Once the core concepts and relationships are defined, the phase may involve expanding the knowledge base with additional details. This can include specifying subtypes or variations of concepts, defining attributes for concepts, and capturing more nuanced relationships with the AEC domain.
   - *Alignment with AEC Standards:* To ensure the ontology's relevance and utility, it is aligned with existing AEC standards, regulations, ontologies, and industry-specific terminology. This alignment helps bridge the gap between knowledge representation and real-world AEC practices.

- *Iterative Process:* The specification of concepts and relationships is often an iterative process, refining the ontology and its associated knowledge base based on feedback and insights gathered from experts and stakeholders within the AEC domain.
- *Integration with Rule Formalisation:* The knowledge captured in this phase serves as the foundation for rule formalisation and the development of rule-specific languages. By having a well-defined knowledge base, the AEC3PO ontology can create rules that are semantically rich, precise, and aligned with industry standards and regulations.

4. **Ontology population:** The ontology population step involves the process of creating instances for the ontology from real data based on specific use cases or application domains [34]. In the context of the ACCORD project, the ontology population is primarily driven by real data from the use cases presented in WP5 of the demo countries. AEC3PO was validated against these use cases to ensure it accurately represents the domain and can effectively support the intended applications. Expert feedback was sought from domain experts and stakeholders involved in the use cases to gain their insights into refining the ontology and ensure it aligns with the practical needs of the project. More details on these use cases are presented in Section 3.4.

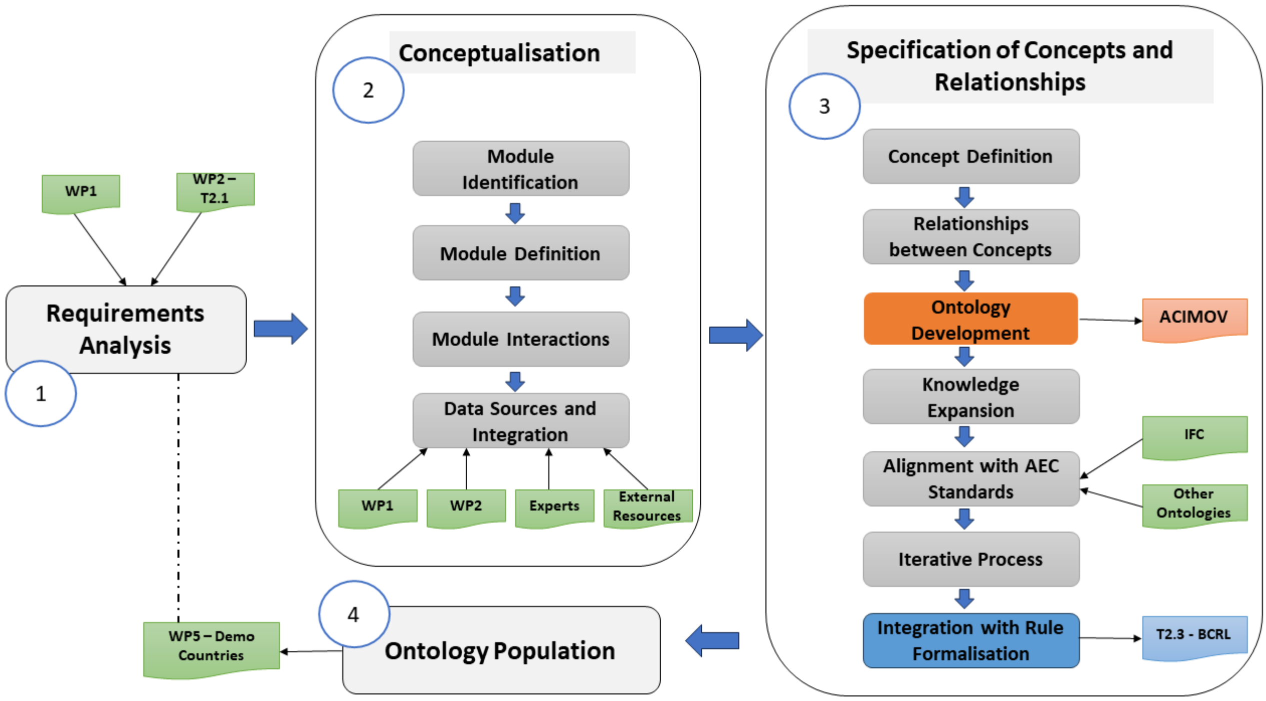


Figure 3 Overview of the main phases of AEC3PO Development Process

3.3.2 AEC3PO Knowledge Acquisition

The development of the AEC3PO ontology within the ACCORD project relies on a variety of data sources to ensure the comprehensiveness and accuracy of the ontology. These data sources have been carefully selected to capture the multifaceted aspects of building regulations and compliance-checking practices across different countries. The following are the primary data sources utilised:

1. **Building Regulations**: The Building Regulations of the UK serve as a foundational data source. These documents provide detailed information about building standards, safety

regulations, and construction requirements within the UK. The English translation of the Finnish Regulations is also incorporated to expand the cross-country perspective.

2. **Regulatory Documents**: Various regulatory documents, including government publications, legal texts, and official guidelines, have been consulted, such as the Code for Sustainable Homes, the BREEAM (Building Research Establishment Environmental Assessment Method), and the International Building Code. These documents offer insights into the specific rules and standards that need to be captured by the AEC3PO ontology.
3. **Expert Consultation**: Domain experts and professionals in the AEC field have been consulted for their valuable insights. Their knowledge and expertise play a crucial role in understanding the nuances of building regulations and construction practices.
4. **ACCORD Documentation**: As part of the ACCORD project, documentation and deliverables related to the semantisation of rules and regulatory compliance are considered, essentially D1.1 and D2.1. These materials offer information on the goals, progress, and outcomes of the project, which are essential for the development of AEC3PO.

These diverse data sources collectively provide a rich foundation for the AEC3PO ontology. By leveraging these sources, we ensure that AEC3PO is capable of comprehensively capturing and representing the intricacies of building regulations and compliance in the construction industry.

3.3.3 AEC3PO Development Methodology

The development of the AEC3PO ontology within the ACCORD project follows the Agile and Continuous Integration for Modular Ontologies and Vocabularies (ACIMOV) ontology development methodology [33], which builds upon the SAMOD (Simplified Agile Methodology for Ontology Development) approach [35] and incorporates collaborative development solutions using Git-based tools. This methodology is tailored to facilitate the creation of modular ontologies and emphasises the reuse of reference ontology modules. It encompasses a well-structured sequence of seven steps that span two development cycles, expertly managed by ontology engineers who oversee backlog management. The longer collaborative cycle engages domain experts and end-product owners, gathering their requirements and involving them in the validation process, especially T2.3, which focuses on the rule formalisation language, which needs to use ontology as a main context to create this language.

ACIMOV emphasises a structured approach that involves sprints and regular meetings to ensure efficient ontology versioning. Additionally, one of the key principles of ACIMOV is to automate processes to enhance productivity and reduce manual efforts. It is important to note that the automation scripts proposed by ACIMOV were initially developed for AEC3PO, and we would like to acknowledge the contribution of this task (T2.2) to the development of automation scripts. The utilisation of automation within AEC3PO reflects the commitment to streamlining the ontology development process and enhancing its efficiency. This approach not only aligns with the principles of ACIMOV but also reflects our dedication to delivering a high-quality ontology with improved productivity.

To implement this methodology, we have employed Github as our collaborative software development platform, along with integrated scripts to support the automatic generation of domain-specific ontologies. This approach ensures a systematic and efficient development process for the AEC3PO ontology within the ACCORD project. Figure 4 shows an overview of the ACIMOV ontology development methodology.

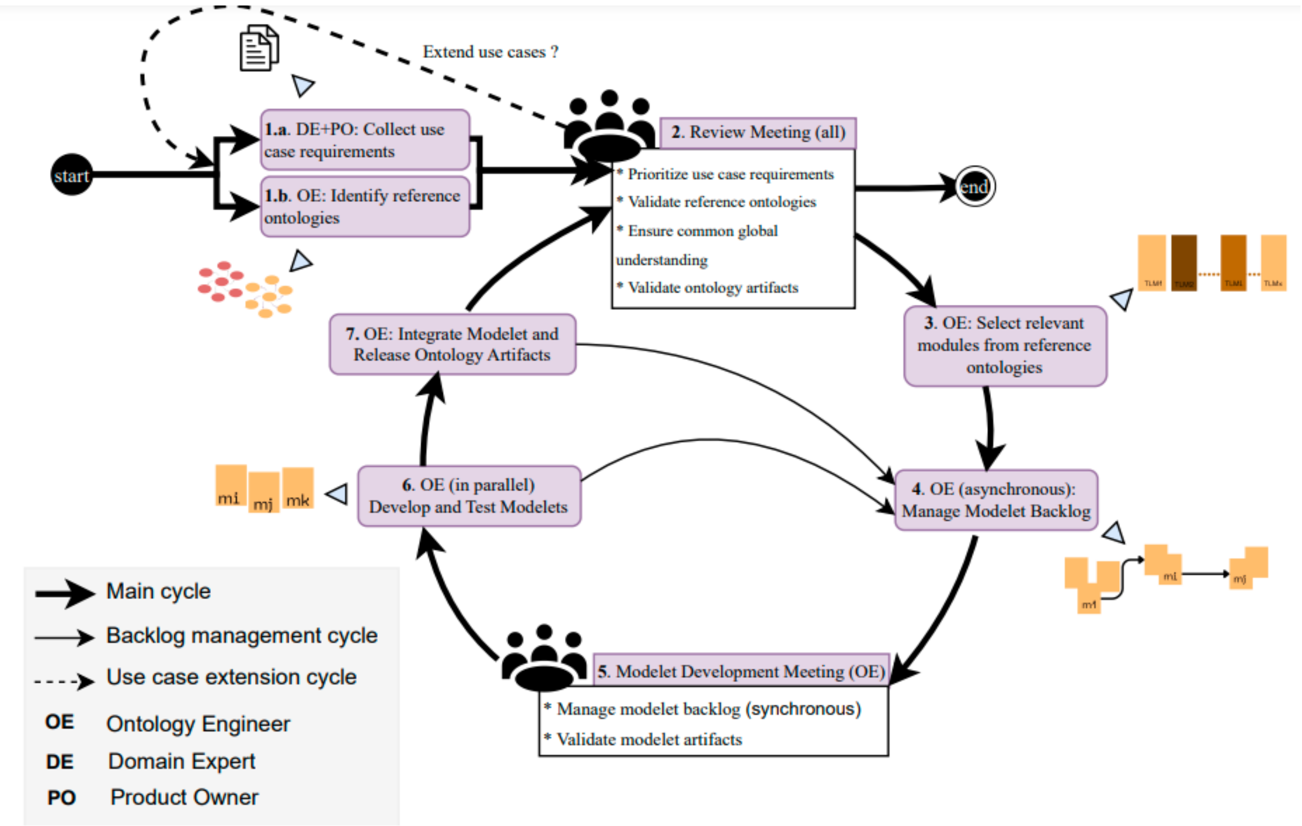


Figure 4 Overview of the ACIMOV Ontology Development Methodology [33]

3.4 AEC3PO Overview

In this section, we delve into a comprehensive overview of AEC3PO, highlighting its core components and emphasising its alignment with existing ontologies. This exploration aims to provide a clear understanding of the ontology structure, modules, and how it integrates with established ontologies in the domain.

3.4.1 Overview of the ontology structure, components, and relationships

AEC3PO aims to model all aspects of compliance and permitting in the AEC domain across different regulatory systems. It is organised into modules comprising classes and properties. Figure 5 shows a generic overview of these modules and their relations. A more fine-granular representation of AEC3PO is available online[2].

[2] w3id.org/lbd/aec3po/

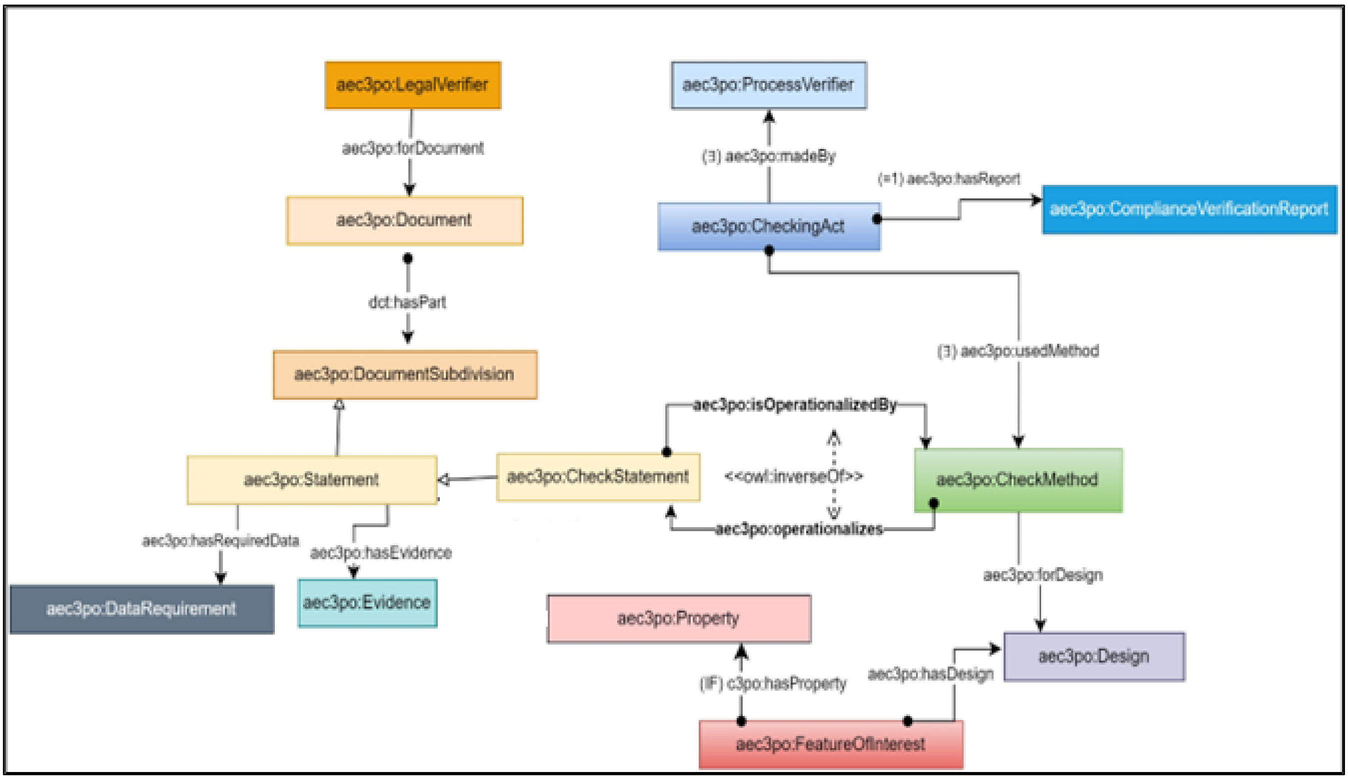


Figure 5 Overview of the Main Modules of AEC3PO

Each module serves a specific purpose and plays a crucial role in the overall structure and functionality of the ontology. We will delve into the details of these modules, their concepts, relationships, and the specific domains they address, shedding light on how they contribute to the broader objectives of AEC3PO. All the AEC3PO modules will be detailed in the following.

- **Module Document:** The AEC3PO *Document* module describes building-compliance related documents, their subdivisions, down to statements and tagged strings or figures. Document subdivisions can be described as literals, linked to their first part and to their immediately following subdivision. This enables roundtripping from a document to its AEC3PO description and back. An overview of the module *Document* is depicted in Figure 6.
  - **Classes:** *Document, DocumentSubdivision, Drawing, Image.*
  - **Properties:** *forDocument, hasFirstSubdivision, hasNextSubdivision, hasSubdivision, requiresDocument, usesDocument.*
  - **Documentation Link:** [*Module Document*]()
  - **Turtle Source:** [*document.ttl*]()

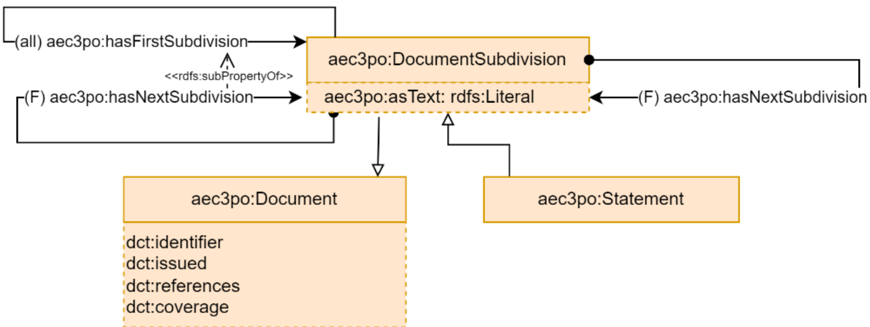


Figure 6 Overview of the AEC3PO Document Module

- **Module Statement:** The AEC3PO Statement module describes requirements stated in a building compliance-related document. Statements can be described as literals using *asText* property. They can be classified into different classes such as *DefinitionStatement, CheckStatement, CheckListStatement* , etc. An overview of the module AEC3PO Statement is depicted in Figure 7.
  - **Classes:** *Statement, CheckStatement, CheckListStatement, AndCheckStatement, ORCheckStatement, DefinitionStatement, BooleanCheckStatement, NumericalCheckStatement, CategoryCheckStatement, CertificateCheckStatement, HumanEvaluatedCheckStatement, NotCheckStatement.*
  - **Properties:** *definitionOf, hasDefinition, hasEvaluator, hasEvidence, hasInlinePart.*
  - **Documentation Link:** *Module Statement*
  - **Turtle Source:** statement.ttl

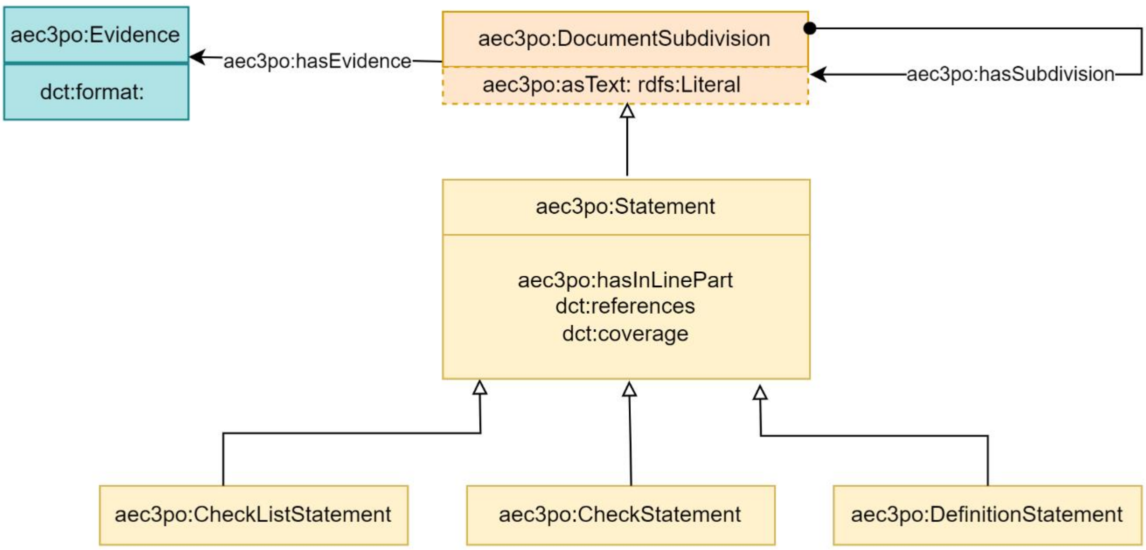


Figure 7 Overview of the AEC3PO Statement Module

- **Module RASEStatement:** The AEC3PO *RASEStatement* module describes statements decomposed following the Requirement Application Selection and Exception (RASE) methodology. An overview of the module RASEStatement is depicted in Figure 8.
  - **Classes:** *RequirementStatement, ApplicationStatement, SelectionStatement, ExceptionStatement*
  - **Properties:** *requires, appliesTo, except, selects*
  - **Turtle Source:** *rase_statement.ttl*

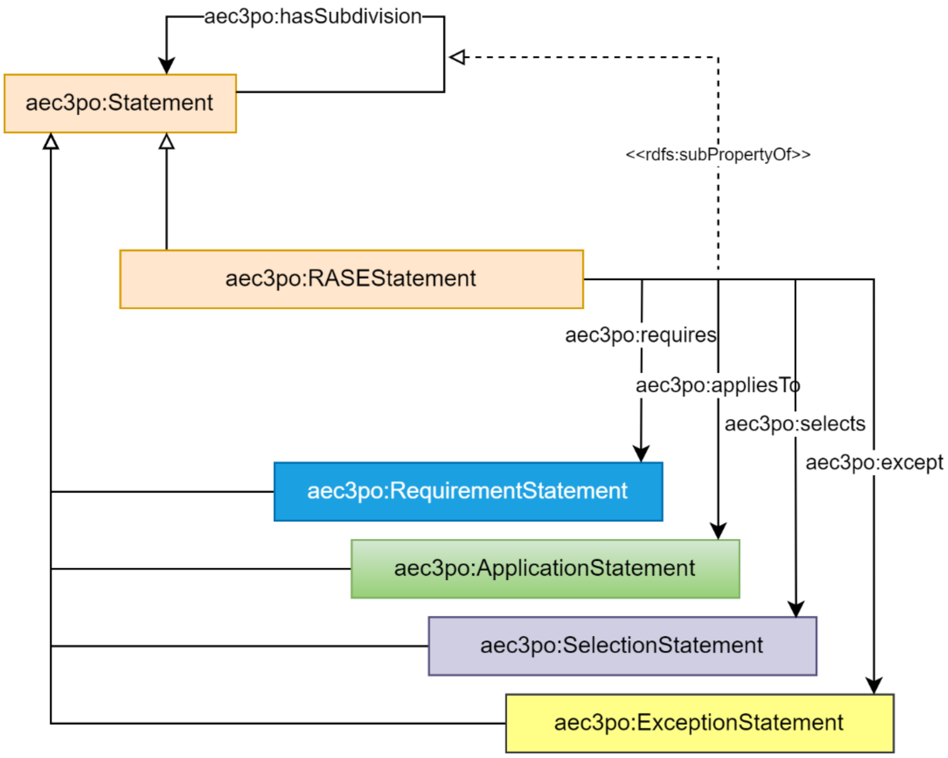


Figure 8 Overview of the AEC3PO RASE_Statement Module

- **Module DataRequirement:** The AEC3PO *DataRequirement* module describes all data requirements that are dictated from the statement. An overview of the module *DataRequirement* is depicted in Figure 9.
  - **Classes:** *DataRequirement, IDS*
  - **Properties:** *hasRequiredData*
  - **Documentation Link:** *Module DataRequirement.*
  - **Turtle Source:** data_requirement.ttl

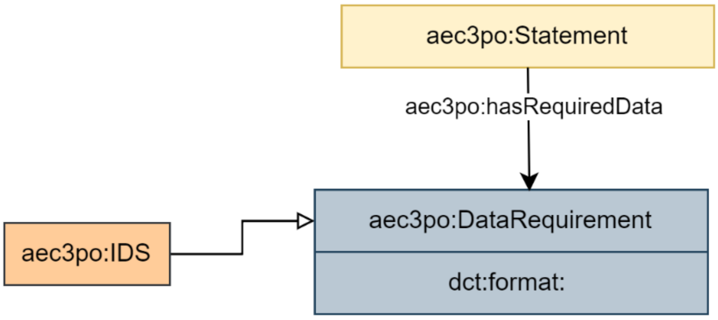


Figure 9 Overview of the AEC3PO DataRequirement Module

- **Module Evidence:** The AEC3PO Evidence module describes the evidence that an actor in the compliance and permitting process needs to provide in order to prove that the requirements derived from a Statement have been met. The evidence can be an image, a drawing or a model. An overview of the module *Evidence* is depicted in Figure 10.
  - **Classes:** *Evidence*

- **Properties:** *hasFormat*
- **Documentation Link:** Evidence Module
- **Turtle Source:** evidence.ttl

Figure 10 Overview of the AEC3PO Evidence Module

- **Module CheckMethod:** Check Method is a piece of information that operationalises check statements in documents, usually executed to control the conformance of some entity. A Check Method is reusable and may be executed several times. An overview of the module *CheckMethod* is depicted in Figure 11.
  - **Classes:** *CheckMethod, BooleanCheckMethod, CertificateCheckMethod, CategoryCheckMethod, ComponentCheckMethod, CompositeCheckMethod, DeclarativeCheckMethod, CheckForConcept, FunctionCheckMethod, NumericalCheckMethod, JenaCheckMethod, ProceduralCheckMethod, SHACLCheckMethod, SWRLCheckMethod.*
  - **Properties:** *hasTarget, hasUnit, isOperationalizedBy, operationalizes.*
  - **Documentation Link:** CheckMethod Module
  - **Turtle Source:** check_method.ttl

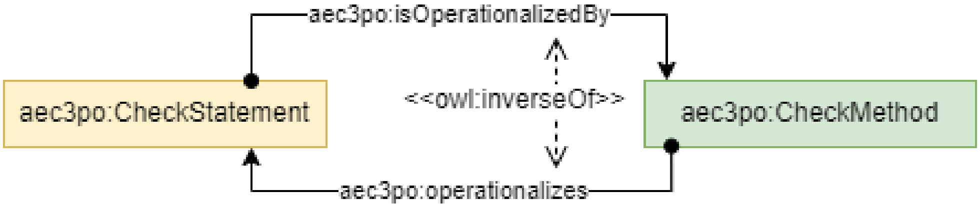


Figure 11 Overview of the AEC3PO CheckMethod Module

- **Module FeatureOfInterest:** The AEC3PO *Feature of Interest* module describes an entity (feature) of a site, building, or piece of infrastructure that is of interest for some purpose. Typically, this will be a building component that needs to be compliant with regulations or documented in the permitting process. The module describes both whole objects (*FeatureOfInterest*) and their aspects (Properties). Properties can hold a text description in string format or may have some quantity kind and are valued. An overview of the module *Feature_Of_Interest* is depicted in Figure 12.
  - **Classes:** *FeatureOfInterest, Property.*
  - **Properties:** *hasProperty.*
  - **Documentation Link:** *Feature_Of_Interest Module*
  - **Turtle Source:** *feature_of_interest.ttl*

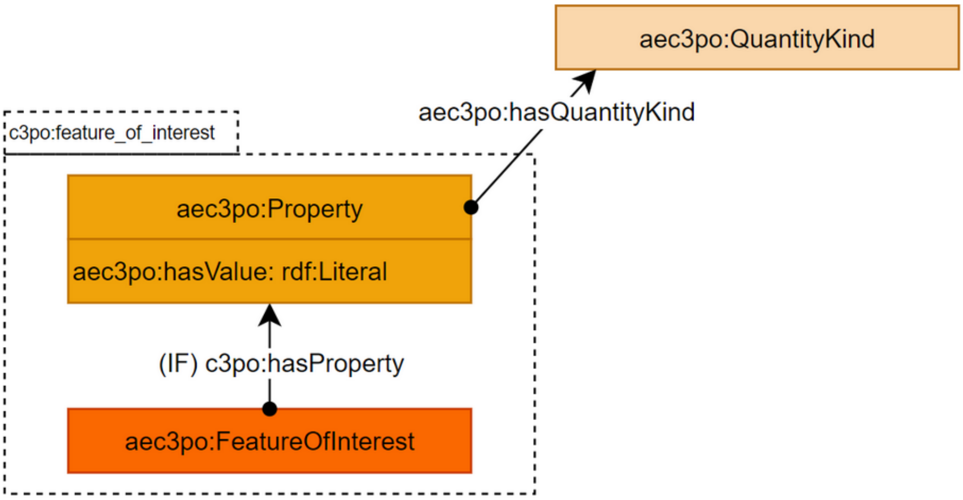


Figure 12 Overview of the AEC3PO Feature of Interest Module

- ***Module CheckingAct:*** The AEC3PO Checking Act module describes the act of checking some entities for something and generating a compliance verification report. An overview of the module *CheckingAct* is depicted in Figure 13.
- Classes: *CheckingAct.*
- **Properties:** *checks, hasReport, madeBy, usedMethod.*
- **Documentation Link:** [CheckingAct Module]
- **Turtle Source:** [checking_act.ttl]

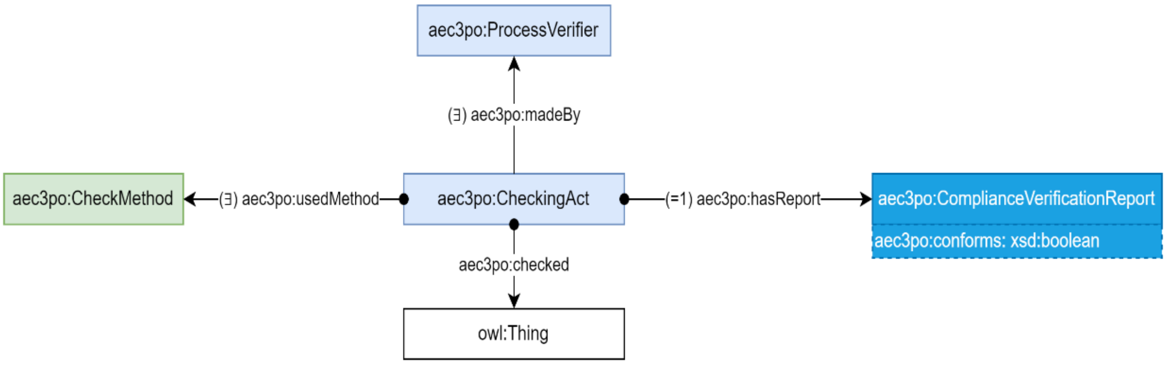


Figure 13 Overview of the AEC3PO Checking Act Module

- **Module ComplianceVerificationReport:** The AEC3PO Compliance Verification Report module describes the results of some "*aec3po:ProcessVerifier*" checking some entity via a *"aec3po:CheckingAct"*. An overview of the module *ComplianceVerificationReport* is depicted in Figure 14.
  - **Classes:** *ComplianceVerificationReport, Severity, ValidationResult, Result.*
  - **Properties:** *focus, result Severity, conforms, message.*
  - **Documentation Link:** [Compliance_Verification_Report Module]
  - **Turtle Source:** [compliance_verification_report.ttl]

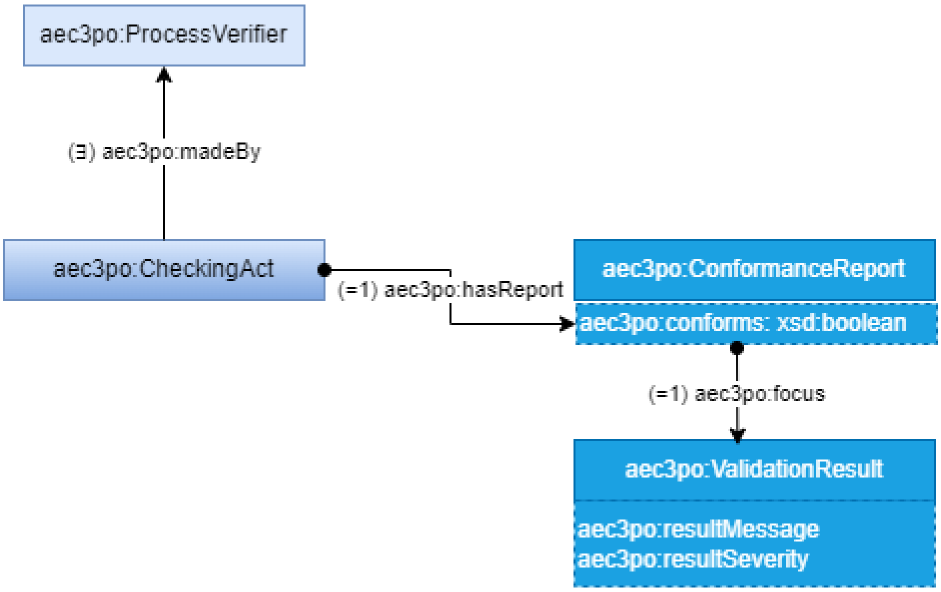


Figure 14 Overview of the AEC3PO Compliance Verification Report Module

- **Module Design:** The AEC3PO *Design* module describes some design features of interest in terms of structure, geometry, and function. An overview of the module *Design* is depicted in Figure 15.
  - **Classes:** *Design, PropertyDesign.*
  - **Properties:** *hasDesign.*
  - **Documentation Link:** Design Module
  - **Turtle Source:** design.ttl

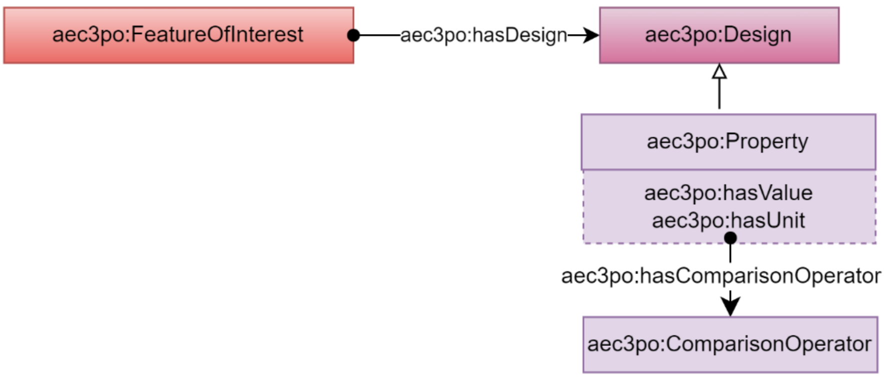


Figure 15 Overview of the AEC3PO Design Module

- **Module Model:** The AEC3PO *Model* module provides a description of the metadata of BIM models. A BIM model is a digital representation of a building or infrastructure project that includes both graphical and non-graphical information. An overview of the module *Design* is depicted in Figure 16.
  - **Classes:** *Model, Phase, Element, Classification, Dimension*
  - **Properties:** *hasBuildingPhase, hasDimension, hasElementPhase, hasClassification, location, locationCoverage.*
  - **Documentation Link:** *Model Module*
  - **Turtle Source:** *model.ttl*

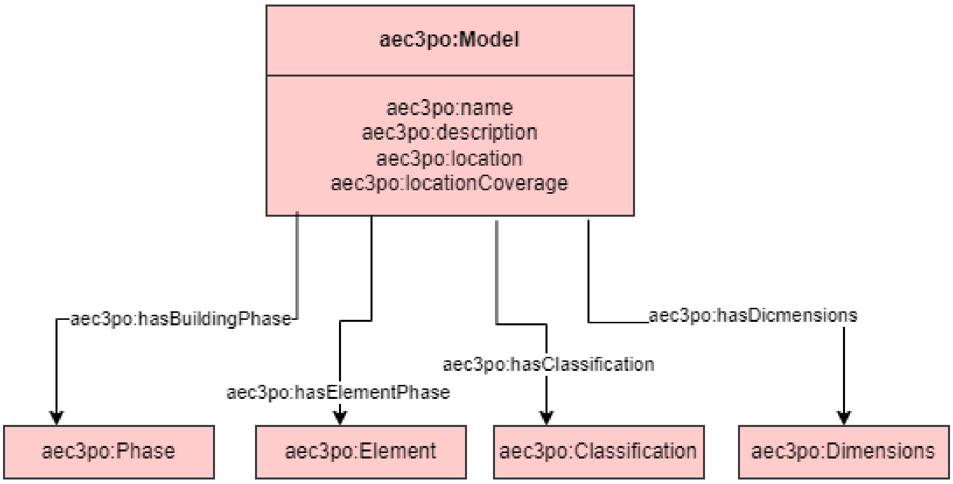


Figure 16 Overview of the AEC3PO Model Module

- **Module LegalVerifier:** The AEC3PO *Legal Verifier* module describes actors of the compliance and permitting process that have the legal capacity to verify that a specific statement of a compliance document has been met in a satisfactory manner. The legal verifier can be either *state verifier* or *private verifier*. An overview of the module *LegalVerifier* is depicted in Figure 17.
- Classes: *LegalVerifier, PrivateVerifier, StateVerifier.*
- Properties: *forDocument.*
- *Documentation Link: Legal_Verifier Module*
- *Turtle Source: legal_verifier.ttl*

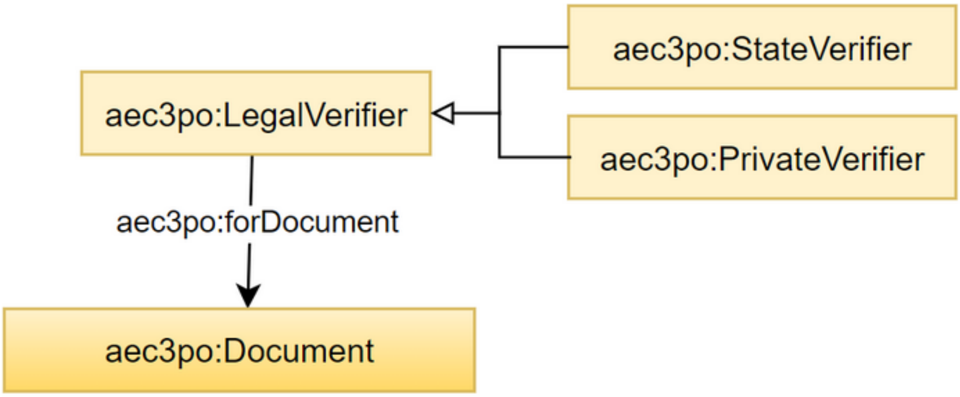


Figure 17 Overview of the AEC3PO Legal Verifier Module

- **Module Table:** The AEC3PO Table module describes tables as representations of data in rows and columns. Tables are described in captions. An overview of the module *Table* is depicted in Figure 18.
    - **Classes:** *Table, Container, Cell, Column, Row.*
    - **Properties:** *contains, IsContainedIn, Caption*
    - **Documentation Link:** Table Module
    - **Turtle Source:** table.ttl

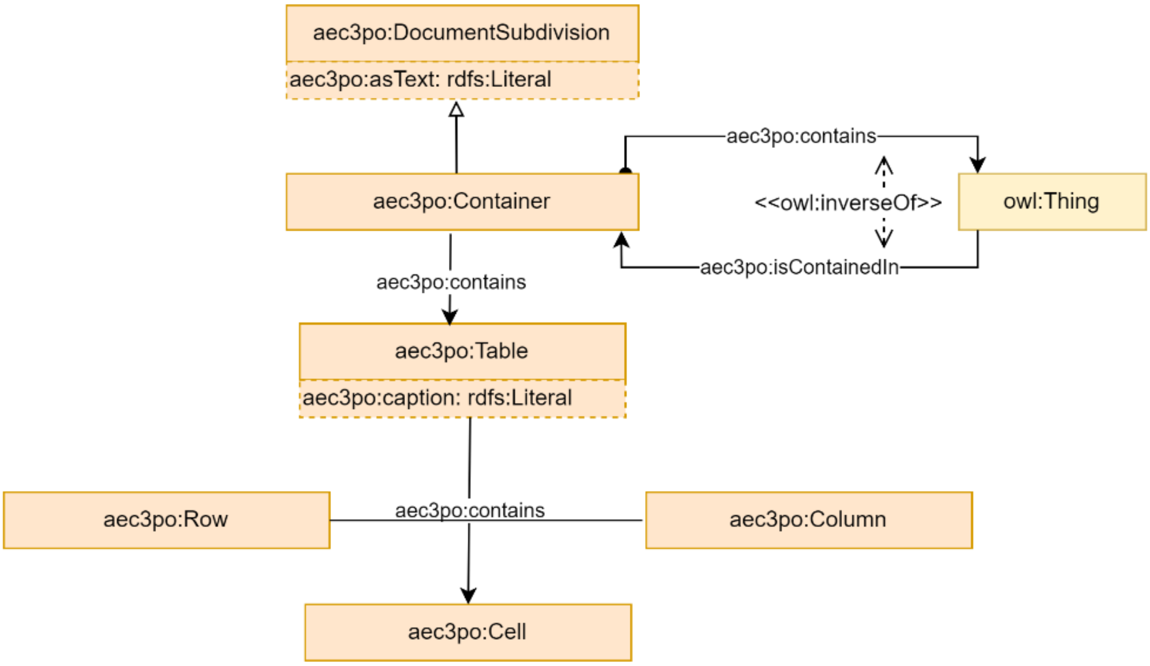


Figure 18 Overview of the AEC3PO Table Module

3.4.2 AEC3PO Alignment

In the development of the AEC3PO ontology, alignment with other ontologies plays a pivotal role in enhancing interoperability and ensuring that AEC3PO seamlessly integrates with existing knowledge resources. The process of ontology alignment involves mapping AEC3PO concepts to those of other established ontologies, creating a common semantic ground for cross-domain knowledge sharing. AEC3PO is strategically aligned with existing building-related ontologies, such as the Industry Foundation Classes (IFC) and other domain-specific ontologies. This alignment facilitates the seamless exchange of information between AEC3PO and other systems used in the architecture, engineering, and construction (AEC) industry. Concepts within AEC3PO are cross-referenced with those from relevant ontologies to ensure consistency and compatibility. Figure 19 below illustrates the alignment of AEC3PO with the imported ontologies. AEC3PO is positioned in the centre, with the imported ontologies represented as separate nodes.

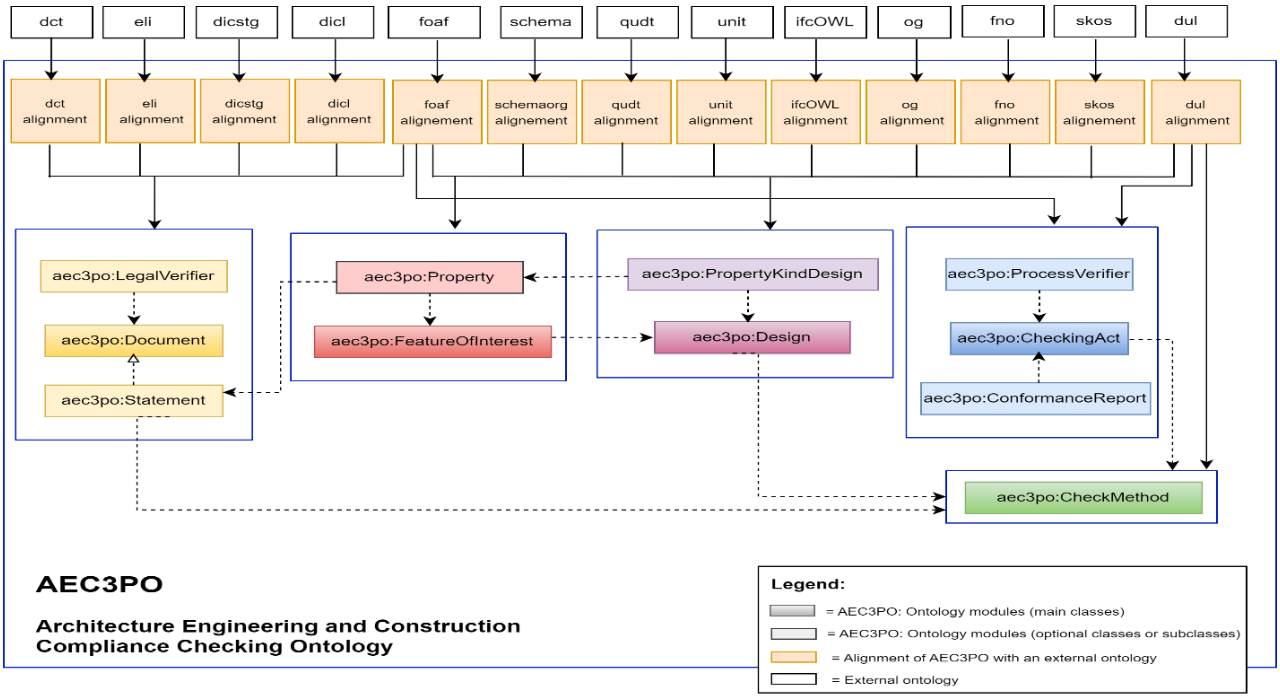


Figure 19 Overview of the AEC3PO Alignments and Reuse of Existing Ontologies

In Table 2, we introduce the aligned ontologies, providing a comprehensive description of each ontology and its specific usage within the AEC3PO framework.

Ontology	**Prefix**	**Description and Usage**
DCMI Metadata Terms	*:dct*	The *Dublin Core Terms (DCT)* ontology is used within the AEC3PO ontology to provide a standardised framework for describing and managing metadata related to documents and other resources in the construction compliance and permitting context.
European Legislation Identifier (ELI)	*:eli*	The *European Legislation Identifier (ELI)* ontology provides a standardised framework for referencing and managing legal and legislative information related to documents, regulations, and other legal entities within the construction compliance and permitting context.
Stages	*:dicstg*	The *Digital Construction Stages* vocabulary is used to define product lifecycle stage frameworks and their specific stages as individuals according to some standards like BS_EN_16310, HOAI, ISO_22263, RIBA.
LifeCycle	*:dicl*	The *Digital Construction LifeCycle* vocabulary is used to define the evolution of information through Level of Development (LoD) and over the construction lifecycle.
Friend of a Friend (FOAF)	*:foaf*	The *Friend of a Friend (FOAF)* ontology is used within the AEC3PO ontology to define agents and organisations such as the *Legal Verifier,* the *State Verifier, etc.*
schema.org	*:schema*	The *schema.org* ontology is used to define the BIM model as a 3D Model, and the different formats that evidence might have such as *image, stillImage* (for drawings), etc.

QUDT	*:qudt*	The *Quantities, Units, Dimensions, and Data Types (QUDT)* ontology provides a standardised way to represent quantities, units of measurement, and their relationships. It is used within the AEC3PO ontology to define the quantities and units represented in a Statement or related to a feature of interest.
Unit	*:unit*	The *Unit* Ontology (Unit) is a resource that provides a standardised way to represent units of measurement and their conversions. It is used within the AEC3PO ontology to provide standardised units for the properties and values.
ifcOWL	*:ifcowl*	The *Industry Foundation Classes (IFC)* ontology in OWL (i*fcOWL*) is a ontology for representing building and construction information. It is used to serve as a reference or a source of domain-specific knowledge that complements the information represented in AEC3PO.
Open Graph Protocol	*:og*	The *Open Graph Protocol (OGP)* ontology describes and represents the properties of a web page or resource. It is used within the AEC3PO ontology to define the URLs of the bSDD contexts of properties and features of interest.
Function	*:fno*	The *Function* Ontology is a lightweight ontology designed to represent functions and their relationships in various domains. It is used within the AEC3PO ontology to represent the functional relationships between different components, systems, and elements in the built environment. The function can be related to an implementation, i.g., SPARQL, SHACL – or a microservice.
SKOS	*:skos*	The *Simple Knowledge Organization System (SKOS)* ontology is commonly used to represent and manage controlled vocabularies, taxonomies, and thesauri. Within the context of the AEC3PO ontology, SKOS is used in various ways to enhance the representation and organisation of concepts and terms related to compliance, design, construction, and permitting processes.
DUL (DOLCE + DnS Ultralite)	*:dul*	The *DOLCE + DnS Ultralite (DUL)* ontology, which is an upper-level ontology, is used in the AEC3PO ontology to provide a foundational framework for modelling and representing various concepts and relationships in a more coherent and structured manner, such as the *CheckMethod, qualities, CheckingAct* , etc.

Table 2 AEC3PO Alignments

3.5 Use Cases and Applications

In this section, we showcase the practical utility and real-world applicability of the AEC3PO ontology. By instantiating the ontology with actual case scenarios drawn from the demonstration countries as presented in Work Package 5 (WP5), we aim to provide a comprehensive understanding of how this ontology can be leveraged to address complex compliance checking challenges within the AEC industry. This instantiation process involves several distinct steps, and we use the Spanish use case as an illustrative example to walk through these stages in Section 3.5.1. To ensure clarity and transparency, we have employed the Turtle format, both in the development of the ontology source code and the presentation of the examples. This approach allows us to delve deep into the intricacies of real-world use cases, emphasising AEC3PO's practical benefits and its potential to semantise compliance checking in the AEC domain.

In the subsequent Section 3.5.2, we will provide concise descriptions of four distinct use cases, each offering a unique perspective on the application of the AEC3PO ontology. These case studies were drawn from the demonstration countries, encompassing two instances from Finland, one from Estonia, and one from the United Kingdom. By exploring such a diverse range of scenarios, we aim to underscore the adaptability and effectiveness of the ontology in addressing compliance checking challenges across international boundaries and different regulatory environments. Through these use cases, we hope to offer a holistic view of the versatile applications of the AEC3PO ontology, fostering a deeper understanding of its potential to drive innovation within the AEC industry.

3.5.1 AEC3PO Instantiation Stages

In order to fully comprehend the intricacies of AEC3PO ontology instantiation, it is essential to outline the various stages involved in this process. This instantiation journey begins with the regulatory documents that serve as the foundation for the rules to be examined. These documents are subsequently defined, establishing their role in compliance checking. Within these regulatory documents, we identify subdivisions and statements that explicitly outline the compliance rules.

To further enhance our understanding, we categorise these statements based on their nature, distinguishing between numerical, categorical, certificate-based, or human-evaluated statements. As we progress, we extract the specific requirements dictated by these statements, expressing the criteria that must be met for compliance.

This rigorous analysis of the regulatory documents and their statements brings us to the core of the ontology instantiation. Moving forward, we delve into real case scenarios that provide practical insights into the feature of interest under evaluation. Here, we uncover the real-world properties of this feature and outline the methods employed for its examination. These methods are attributed to distinct checking acts, which play a pivotal role in the compliance verification process.

In the final stages of the ontology instantiation, we report the outcomes of the compliance checks in the form of Conformance Verification Reports. This meticulous process allows us to encapsulate every aspect of the compliance checking journey, from regulatory documents to real-world applications, resulting in a comprehensive and robust ontology for the AEC industry.

The primary objective of the instantiation of AEC3PO is to capture the entire compliance and permitting checking process rather than solely concentrating on rule formalisation. This approach aims to comprehensively represent the diverse facets of compliance and permitting checks, encompassing regulatory documents, their subdivisions, statements, requirements, real-world properties, examination methods, checking acts, and conformance verification reports.

By adopting this more holistic approach, the instantiation of AEC3PO provides a comprehensive and detailed representation of the compliance checking process within the AEC industry, offering a broader perspective than the rule-centric formalisation of the RASE methodology. Furthermore, it is worth noting that AEC3PO serves as the foundational context within which the rule formalisation methodology is applied. This ontology provides the structural framework for understanding, categorising, and capturing the entire compliance and permitting checking process, enabling the effective application of various rule formalisation methodologies within the domain of the AEC industry, including RASE methodology.

To concretise the instantiation stages and provide a practical illustration of the process, we will begin by introducing the Spanish use case. This particular case involves the evaluation of a cultural centre, as described in the subsequent section. With this introduction in mind, we will proceed to demonstrate the different stages of AEC3PO instantiation by providing detailed code snippets from the accompanying Turtle file, which will offer a clear and practical insight into the instantiation process for the Spanish use case. This will provide a comprehensive understanding of how AEC3PO operates within real-world scenarios and facilitate the seamless conformance verification of building models against regulatory documents.

3.5.1.1 Description of the Spanish Use Case

This Spanish use case (use case 1) involves the evaluation of a cultural centre, as depicted in Figure 20.

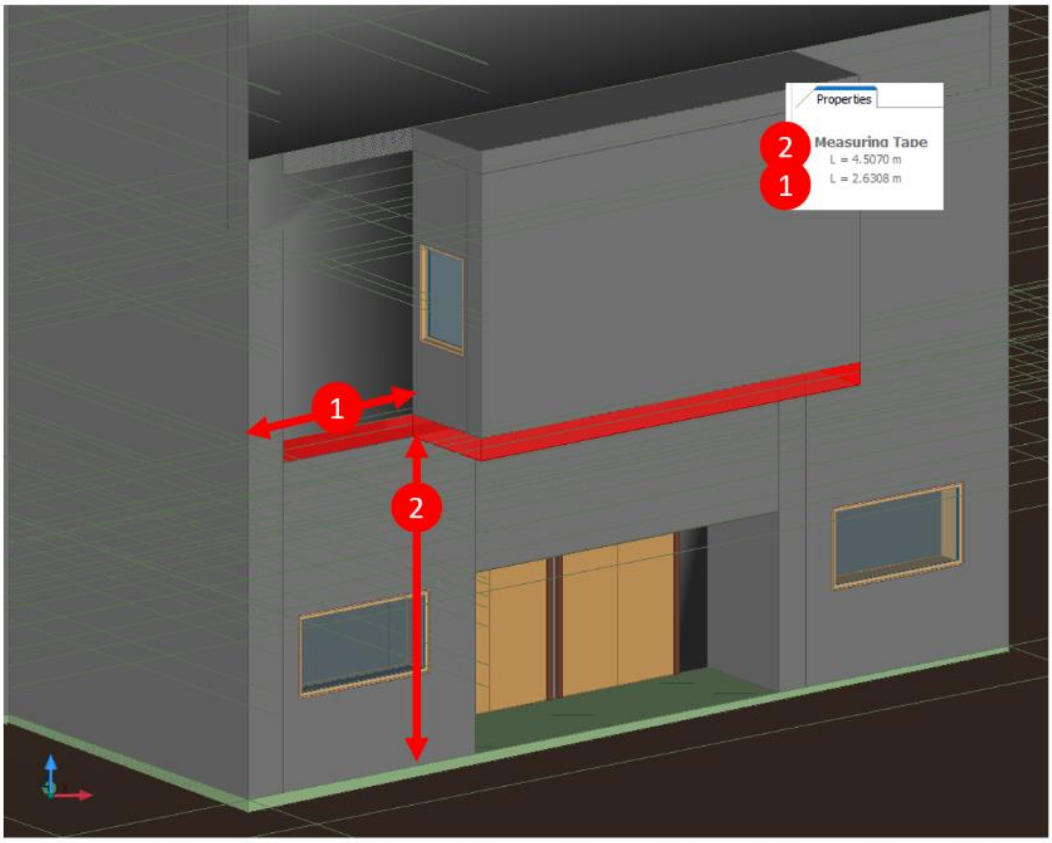


Figure 20 Overview of the cultural centre treated in the Spanish Use Case 1

The assessment ensures compliance with the regulations specified in the Municipal Urban Planning Plan (POUM) document. Notably, the POUM document, which was officially approved by the Barcelona Territorial Planning Commission on July 13th, 2005, serves as the primary reference for this compliance check. Specifically, two key statements from the POUM document have been selected to address the necessary checks for this building model. These two statements inherently represent the rules associated with the base Offset and the party-wall distance of the cantilever in the context of this specific building. These two statements are respectively presented in *Part2/Chapter1/Section2* and *Part2/Chapter2/Section2* from the POUM document, and the English translation of the respective clauses are shown below in Figure 21.

"The minimum free height of cantilevers over public spaces will be 3.20 metres for street with a width of 8.00 metres or less and 3.50 metres in all other cases." – POUM, Part2/Chapter1/Section2

" All cantilever must be separated by at least 1 meter from the line of the partywall. " – POUM, Part2/Chapter2/Section2

Figure 21 The two statements from POUM document to be checked for the Spanish Use Case 1

3.5.1.2 Illustration of the Instantiation Stages

As detailed earlier, the initial stage of the instantiation process involves the creation of an instance representing the regulatory document that contains the rules. To execute this stage, we employ the "*Document"* class within AEC3PO (Figure 22). The document instance is further characterised by its metadata, which is represented using concepts from the Dublin Core Terms (DCT) ontology, which we have effectively incorporated for this purpose. These metadata attributes help define and contextualise the regulatory document within the AEC3PO ontology.

```
# Instantiate the POUM Document as a Document
ex:POUM_Doc a aec3po:Document ;
    dct:identifier <https://www.ajmalgrat.cat/informacio-oficial/urbanisme-i-obra-publica/poum> ;
    dct:issued "2005-07-13"^^xsd:date ;
    dct:references <reference_here> ;
    dct:coverage aec3po:Spain .
```

Figure 22 AEC3PO Instantiation Stage 1 – Example of Document Instantiation

Once the regulatory document is defined within the AEC3PO ontology, the next stage is to define its subdivisions, which correspond to sections or chapters of the document that contain the specific rules to be checked with the ontology class "*DocumentSubdivision"*. These subdivisions are linked to the initial document instance using the "*hasPart"* property, as shown in Figure 23. This allows for a hierarchical representation of the document and its various components, facilitating the organisation and categorising rules for further processing.

```
# Instantiate the first DocumentSubdivision that contains the rule to be checked
ex:POUM_DocSubdivision_Part2_CH1_S2 a aec3po:DocumentSubdivision ;
    dct:identifier "POUM/Part_2/Chapter_1/Section_2" ;
    dct:title "Part2_CH1_S2" .

# Instantiate the second DocumentSubdivision that contains the rule to be checked
ex:POUM_DocSubdivision_Part2_CH2_S2 a aec3po:DocumentSubdivision ;
    dct:identifier "POUM/Part_2/Chapter_2/Section_2" ;
    dct:title "Part2_CH2_S2" .

# Link the DocumentSubdivision to the Document using hasPart property
ex:POUM_Doc dct:hasPart  ex:POUM_DocSubdivision_Part2_CH1_S2 , ex:POUM_DocSubdivision_Part2_CH2_S2 .
```

Figure 23 AEC3PO Instantiation Stage 2 – Example of DocumentSubdivision Instantiation

In the subsequent stage of the instantiation process, we define and instantiate the individual statements using the class *"Statement"* (Figure 24). Each statement is represented by its content, which is defined using the "*asText"* property. These statements are then linked to their respective document subdivisions through the "*hasPart"* property, establishing a clear and organised structure for referencing the rules to be checked within the document subdivisions.

```
# Instantiate the first statement
ex:statement_54 a aec3po:Statement ;
    dct:identifier "POUM/Part_2/Chapter_1/Section_2/Clause_54.8" ;
    dct:title "54.8" ;
    aec3po:asText "The minimum free height of cantilevers over public spaces will be 3.20 metres for street with a width of 8.00 metres or

# Link the first statements to the DocumentSubdivision using hasPart property
ex:POUM_DocSubdivision_Part2_CH1_S2 dct:hasPart  ex:statement_54 .

# Instantiate the second statement
ex:statement_63 a aec3po:Statement ;
    dct:identifier "POUM/Part_2/Chapter_1/Section_2/Clause_63.3" ;
    dct:title "63.3" ;
    aec3po:asText "All cantilever must be separated by at least 1 meter from the line of the partywall." .

# Link the second statement to the DocumentSubdivision using hasPart property
ex:POUM_DocSubdivision_Part2_CH2_S2 dct:hasPart  ex:statement_63  .
```

Figure 24 AEC3PO Instantiation Stage 3 – Example of Statement Instantiation

In the following phase of the instantiation process, we take a more granular approach by categorising and defining the statements according to their specific attributes and characteristics (Figure 25). Each statement is categorised based on its nature, which may include designations such as numerical (*"NumericalCheckStatement"*), human-evaluated (*"HumanEvaluatedCheckStatement")*, categorical (*"CategoryCheckStatement"*), or certificate-based (*"CertificateCheckStatement"*), among others. This categorisation helps to provide a comprehensive representation of the various types of rules and regulations within the ontology, allowing for a more detailed analysis and verification of each statement and the corresponding check method. In this specific case of the Spanish use case, all statements fall under the category of numerical statements, meaning they are quantifiable and relate to numerical values that need to be checked for compliance.

```
# Instantiate the streetBaseOffset NumericalCheckStatement as a subClassOf CheckStatement
ex:cantiliver_numericalCheckStatement_streetBaseOffset rdf:type aec3po:NumericalCheckStatement .

# Instantiate the otherBaseOffset NumericalCheckStatement as a subClassOf CheckStatement
ex:cantiliver_numericalCheckStatement_otherBaseOffset rdf:type aec3po:NumericalCheckStatement .

# Instantiate the partywall distance NumericalCheckStatement as a subClassOf CheckStatement
ex:cantiliver_numericalCheckStatement_partywall_distance rdf:type aec3po:NumericalCheckStatement .
```

Figure 25 AEC3PO Instantiation Stage 4 – Example of CheckStatement Instantiation

The next stage of instantiation involves defining the data related to the statement. This process starts with the "*Feature of Interest"*, which refers to an entity, component, or aspect of a building or infrastructure that is the subject of interest for compliance or permitting checks (Figure 26). The "*Feature of Interest"* module encompasses both whole objects and their specific aspects, referred to as "*Properties"*. These properties are characterised by their quantity kind and are associated with specific values. When applicable, we aim to link these properties to corresponding IFC objects.

In the context of the Spanish use case, the "*Feature of Interest"* is identified as the "*Cantilever"*, which corresponds to the *IfcSlab* entity. This connection between the "*Feature of Interest"* and the corresponding IFC object helps to establish a clear mapping of the feature of interest to the actual building component. This instantiation process ensures that the ontology accurately represents real-world objects and their relevant properties for compliance and permitting checks.

```
# Instantiate the cantilever as a FeatureOfInterest and link it to ifcSlab.
ex:cantilever a aec3po:FeatureOfInterest ;
    rdf:type ifcOWL:ifcSlab .
```

Figure 26 AEC3PO Instantiation Stage 5 – Example of FeatureOfInterest Instantiation

Following the definition of the "*Feature of Interest"* and its identification in the ontology, the next step is to specify its properties as guided by the statements within the regulatory document. Each statement corresponds to one or more particular properties of the "*Feature of Interest"*. These properties are associated with their respective statements to ensure accurate representation and alignment with the compliance and permitting checks. The relationship between the "Feature of Interest" and its properties is established using the property "*hasRequiredData"* (Figure 27).

This approach allows for a clear and structured connection between the "*Feature of Interest"* and the properties that are essential for compliance and permitting checks. AEC3PO ensures that all relevant data and properties are correctly attributed to the feature of interest, enabling precise and automated rule checking.

```
# Define the properties of the cantilever and constraints based on the Statement
ex:streetBaseOffset a aec3po:Property ;
    aec3po:hasValue 3.2 ;
    qudt:hasUnit unit:M ;
    aec3po:hasComparator aec3po:CheckMethodComparisonOperator-eg .

ex:othersBaseOffset a aec3po:Property ;
    aec3po:hasValue 3.5 ;
    qudt:hasUnit unit:M ;
    aec3po:hasComparator aec3po:CheckMethodComparisonOperator-eg .

ex:partywall_distance a aec3po:Property ;
    aec3po:hasValue 1 ;
    qudt:hasUnit unit:M ;
    aec3po:hasComparator aec3po:CheckMethodComparisonOperator-ge .

ex:statement_54 aec3po:hasRequiredData ex:streetBaseOffset .
ex:statement_54 aec3po:hasRequiredData ex:othersBaseOffset .
ex:statement_63 aec3po:hasRequiredData ex:partywall_distance .
```

Figure 27 AEC3PO Instantiation Stage 6 – Example of Property Instantiation

With the properties (*"Property"*) and the "*Feature of Interest"* defined, the next crucial step in the ontology instantiation process is to specify the check methods ("*Check Method"*) that will be employed to perform the actual checks. These check methods are informed by the specific data and requirements of the given use case. Depending on the nature of the checks, various methods may be utilised, such as microservices, numerical algorithms, certificate-based assessments, and more. Each "*Check Method"* is associated with the corresponding "*CheckStatement"* it will be responsible for operationalising. This linkage is established through the property "*operationalizes"*, ensuring that the appropriate method is applied to verify the compliance or permitting criteria outlined in the statements. The ontology's capacity to accommodate a range of check methods provides flexibility and adaptability to diverse use cases, enabling precise and customised rule-checking processes (Figure 28). This stage of ontology instantiation is critical for aligning the chosen methods with the regulatory requirements and the specific properties of the feature of interest, ultimately facilitating accurate and efficient compliance and permitting checks.

In the context of the Spanish use case, the selected check method is microservices, as reflected by the instantiation of the class "*ProceduralCheckMethod"*. By using microservices, a procedural approach is followed to ensure that the necessary checks are carried out in a systematic and rigorous manner. AEC3PO accommodates such specific methodological preferences, exemplifying its adaptability and capacity to represent diverse compliance and permitting processes.

```
# Define and Instantiate all the CheckMethod(s) that will operationalize the Statement defined earlier.

ex:cantiliver_numericalCheckMethod_streetBaseOffset rdf:type aec3po:ProcedurialCheck ;
  aec3po:operationalizes ex:cantiliver_numericalCheckStatement_streetBaseOffset .
  #asText "The minimum free height of cantilevers over public spaces will be 3.20 metres for street with

ex:cantiliver_numericalCheckMethod_otherBaseOffset  rdf:type aec3po:ProcedurialCheck ;
aec3po:operationalizes ex:cantiliver_numericalCheckStatement_otherBaseOffset .
    #asText "3.50 metres in all other cases."

ex:cantiliver_numericalCheckMethod_partywall_distance  rdf:type aec3po:ProcedurialCheck ;
 aec3po:operationalizes  ex:cantiliver_numericalCheckStatement_partywall_distance .
#asText "All cantilever must be separated by at least 1 meter from the line of the partywall."
```

Figure 28 AEC3PO Instantiation Stage 7 – Example of CheckMethod Instantiation

In the ontology instantiation process of the Spanish use case, when considering the *base-offset*, we should note that two sub-checks are associated with it, which are linked using a logical OR operator. Additionally, in combination with the *party-wall* check, they represent an AND operation. To accommodate this logic, AEC3PO defines both *OrCheck* and *AndCheck* classes within the ontology. These classes are used to represent the logical conditions within the compliance checks. Specifically, *OrCheck* is utilised to denote an OR operation, while *AndCheck* signifies an AND operation. These logical operations are employed to articulate the relationships between the sub-checks. Subsequently, these logical checks, whether OR or AND, are linked to the respective sub-checks using the property "*hasSubCheck"* (Figure 29). This modelling approach ensures that the compliance checks are accurately represented in the ontology, capturing the complex logical structures that may be required for thorough rule evaluation.

```
# Instantiate the ORCheck for the baseOffset
ex:cantiliver_ORCheck_baseOffSet a aec3po:OrCheck ;
    aec3po:hasSubCheck ex:cantiliver_numericalCheckMethod_streetBaseOffset , ex:cantiliver_numericalCheckMethod_otherBaseOffset .

# Instantiate the AndCheck related to the two statements, one that checks the baseOffset and the other checks the distance to party wall
ex:cantiliver_AndCheck a aec3po:AndCheck ;
    aec3po:hasSubCheck ex:cantiliver_ORCheck_baseOffSet , ex:cantiliver_numericalCheckMethod_partywall_distance .
```

Figure 29 AEC3PO Instantiation Stage 8 – Example of OrCheck and AndCheck Instantiation

By employing these logical check classes and associations, AEC3PO enhances its capability to handle intricate compliance verification scenarios effectively and facilitates accurate representation of the compliance checks per the regulations and standards.

It is important to highlight that in the RASE methodology, these explicit logical operations, such as OR and AND, are not explicitly needed since they are implicitly captured through the RASE tagging system. However, AEC3PO adopts a more general and flexible perspective to accommodate a wide range of compliance verification scenarios. This allows AEC3PO to provide a comprehensive representation that can effectively address the diverse and complex logic required by different regulations and standards.

Certainly, in the next stage of ontology instantiation, we proceed to define the real feature of interest and its properties based on the specifics of the actual use case. To achieve this, we utilise the class "*Design"* within AEC3PO to represent the feature of interest, which typically corresponds to a 3D model or design in the construction domain. Additionally, we employ the class *"PropertyDesign"* to define the various properties associated with this design.

Once these properties have been specified, we establish the necessary relationships by linking them to the "*Design"* object using the property "*hasProperty"* (Figure 30). This process ensures that the properties are appropriately associated with the feature of interest, facilitating a comprehensive representation of the real-world scenario within the ontology. The purpose of this example is to showcase the logic of the ontology to semantically model the whole process of compliance and permitting checking starting from a regulatory document. However, it is worth noting that in the future, we will consider utilising some tools to automate this process.

```
# Start describing the "real" Cantilever from the Spanich Demo Case
# Define the design of the cantilever as a 3DModel
ex:cantilever_design a aec3po:Design ;
    rdf:type schema:3DModel .

# Define the design of the properties of the cantilever
ex:design_streetBaseOffset a aec3po:PropertyDesign ;
    aec3po:hasValue 4.5 ;
    qudt:hasUnit unit:M .
# I am assuming this represents 2 in the image.

ex:design_othersBaseOffset a aec3po:PropertyDesign ;
    aec3po:hasValue 2.6 ;
    qudt:hasUnit unit:M .
#I am assuming this represents 1 in the image.

ex:design_partywall_distance a aec3po:PropertyDesign ;
    aec3po:hasValue 1 ;
    qudt:hasUnit unit:M .

#Link all the properties design to the cantilever design using hasProperty property
ex:cantilever_design aec3po:hasProperty ex:design_streetBaseOffset .
ex:cantilever_design aec3po:hasProperty ex:design_othersBaseOffset .
ex:cantilever_design aec3po:hasProperty ex:design_partywall_distance .
```

Figure 30 AEC3PO Instantiation Stage 9 – Example of Design and PropertyDesign Instantiation

In the subsequent phase of ontology instantiation, we proceed to establish the connections between the defined properties of the feature of interest ("*Design*") and the check methods that were previously defined. To achieve this, we utilise the property "*forDesign"* to link each property design to the relevant check methods (Figure 31). By connecting these elements, we ensure that each property is associated with the appropriate check methods, enabling the ontology to represent the relationships between real-world properties and the methods used to verify compliance or perform checks in the given use case. This linkage is crucial for a comprehensive representation of the compliance-checking process within the ontology.

```
# Link all the properties designs to the respective CheckMethods using forDesign property
ex:cantiliver_numericalCheckMethod_streetBaseOffset aec3po:forDesign ex:design_streetBaseOffset .
ex:cantiliver_numericalCheckMethod_otherBaseOffset aec3po:forDesign ex:design_othersBaseOffset .
ex:cantiliver_numericalCheckMethod_partywall_distance aec3po:forDesign ex:design_partywall_distance .
```

Figure 31 AEC3PO Instantiation Stage 10 – Example of forDesign Instantiation

In the final stage of the ontology instantiation process, we proceed to define the Checking Act using the corresponding class *"CheckingAct"* within AEC3PO. The *CheckingAct* is responsible for executing the compliance checks using the previously defined check methods. Each *CheckingAct* is linked to the appropriate check methods through the "*usedMethod"* property. As part of the compliance verification process, AEC3PO defines the "*ComplianceVerificationReport"* to store the results of the checks. These results are represented as boolean values, indicating whether the feature of interest conforms to the regulations or not. The Compliance Verification Report contains the *"conforms"* property, which is used to record and communicate the compliance status. To establish the relationship between the *CheckingAct* and the *ComplianceVerificationReport*, we utilise the "*hasReport"* property (Figure 32). This linkage ensures that the results of the compliance checks are stored and associated with the respective Checking Act within the ontology. This comprehensive approach allows AEC3PO to effectively model the entire process of compliance checking and verification, from defining the checks to recording the outcomes in a structured manner.

```
# Conduct the checking act and generate a conformance report.
ex:cantiliver_checkingAct a aec3po:CheckingAct .

# Link the checkingAct to all the CheckMethods using usedMethod property
ex:cantiliver_checkingAct aec3po:usedMethod ex:cantiliver_numericalCheckMethod_streetBaseOffset ,
 ex:cantiliver_numericalCheckMethod_otherBaseOffset , ex:cantiliver_numericalCheckMethod_partywall_distance .

# Instantiate the conformance report
ex:cantiliver_ConformanceReport a aec3po:ConformanceReport ;
    aec3po:conforms true .
```

Figure 32 AEC3PO Instantiation Stage 11 – Example of CheckingAct and ComplianceVerificationReport Instantiation

3.5.2 Demo Countries' Use Cases

The instantiation of AEC3PO is further enriched by considering the specific use cases from different countries. In this section, we introduce the use cases from Finland, Estonia, and the UK, encompassing a wide range of scenarios to ensure the diversity and coverage of ontology classes. Each use case provides a unique perspective on the application of AEC3PO, reflecting the complexities and variations encountered in real-world compliance and permitting processes.

These diverse use cases serve as valuable demonstrations of how AEC3PO can effectively capture and represent the intricacies of compliance checks across different regulatory environments. In the subsequent sections, we will briefly describe each use case, providing an overview of the context and the specific scenarios it addresses.

3.5.2.1 Estonia Use Case – Operational Map

In this instance, we have performed the ontology instantiation using the Estonian Use Case, specifically, Estonia demo case 1, which focuses on fire safety compliance. The compliance checks are conducted against the Fire Safety Requirements specified in the Estonian building regulations, which were issued on March 1st, 2021. Within this context, we have chosen two specific statements from the regulations to serve as the basis for our compliance checks.

These two selected statements correspond to two distinct rules related to the operational mapping of the building. These two statements are respectively presented in *section 52, clauses 5 and 7.* More specifically, the first statement, identified as "*clause 52.5"*, falls under the category of *CategoryCheckStatement* and is defined as follows: "*ex:EE_FireSafety_Doc_Section_52_Clause5 a aec3po:CategoryCheckStatement".* The second statement, referred to as "*clause 52.7"*, represents a *CheckListStatement* and is defined as follows: "*ex:EE_FireSafety_Doc_Section_52_Clause7 a aec3po:CheckListStatement".* This *CheckListStatement* encompasses a list of BooleanCheckStatements, defined as instances of the *aec3po:BooleanCheckStatement* class and linked to the *CheckListStatement* through the property "*aec3po:hasSubCheck"*. Furthermore, the Operational Map has been instantiated as an instance of the *aec3po:FeatureOfInterest* class, with its various properties represented using the *aec3po:Property* class. This example illustrates the instantiation of different categories of check statements within the AEC3PO ontology, including *CategoryCheckStatement, BooleanCheckStatement, and CheckListStatement*.

More detailed information regarding this example and its implementation is available on Github [3].

3.5.2.2 Finland Use Case 1 – Accessibility

This example, provided in the file "*FI-accessibility-AEC3PO.ttl"*, demonstrates the instantiation of the AEC3PO ontology using Turtle syntax with the Finnish demo case 1, focusing on compliance checking of accessibility. Specifically, it represents a check related to ramps. The rules for this check are extracted from Section 2/Subsection 2 of the English translation of the Finnish Accessibility document, derived from the Land Use and Building Act (132/1999) as amended by Act 958/2012.

In this example, a variety of check statements are instantiated, including *NumericalCheckStatement* and *HumanEvaluatedCheckStatement*, to cover different aspects of the compliance check.

3.5.2.3 Finland Use Case 3 – CO2 Emission

The example provided in the file "*FI3-CO2_Emission-AEC3PO.ttl"* showcases the instantiation of the AEC3PO ontology using Turtle syntax with the Finnish demo case 3 related to CO2. This specific example pertains to the Carbon footprint emission check, focusing on compliance with environmental regulations. The rules governing this check are defined in the English translation of the Decree of the Ministry of the Environment on the climate assessment of buildings (Draft 30.9.2022, consultation round). The statements corresponding to these rules are available in an online document[4]. In this example, a wide range of check statements are utilised to address different facets of compliance, including *NumericalCheckStatement*, *BooleanCheckStatement*, *CategoryCheckStatement*, and *CheckListStatement*. This diverse use of check statement classes exemplifies the versatility of AEC3PO in accommodating various compliance requirements.

Notably, this example highlights the adaptability of AEC3PO by demonstrating how it can handle rules presented in a tabular format by using the module *Table*. Real data from the use case have been employed to instantiate the feature of interest, which, in this instance, is the *building*. The use of microservices for the compliance check is also showcased, with the *ProceduralCheckMethod* class employed to define the check method, emphasising the flexibility of the ontology in supporting different operational procedures.

3.5.2.4 Common European Regulations for Structural Engineering (Eurocodes)

The example found in the file *"UK_Structure.ttl" focuses on* a typical strength check found in the Eurocodes, i.e. the European Standards EN 1990 - EN 1999, which provide a common approach for the design of buildings and other civil engineering works. This is the de facto structural engineering design standard in most European countries, including the UK.

[3] https://github.com/Accord-Project/aec3po/tree/main/examples/Estonia

[4] https://vttgroup.sharepoint.com/:w:/r/sites/EU-projectpreparationDigitalpermitsandcompliancecheck/_layouts/15/doc2.aspx?sourcedoc=%7BB135D7E5-FBF8-4AA2-A12D-7F5856EE7A38%7D&file=Use%20Case%20FI3%20extract%20from%20the%20regulations.docx&action=default&mobileredirect=true&cid=495d26c0-9bbb-4fa1-a732-648047ecf92b&wdLOR=c4CB228AF-B53E-4B5F-8938-71C14B46458A

The implemented example is a strength check in compression parallel to the grain, applicable to timber members. This is drawn from the current version of Eurocode 5 (EN 1995-1-1:2004+A2:2014), specifically Clause 6.1.4.

This example introduces another facet of AEC3PO, specifically its capability to handle *EquationStatement*. The clause is represented as an equation within the ontology, emphasising the adaptability of AEC3PO in accommodating various types of statements and compliance checks.

3.6 AEC3PO Evaluation

The AEC3PO ontology underwent a comprehensive evaluation to ensure its quality, usability, and effectiveness. The evaluation process involved four key methodologies.

3.6.1 Statistical Analysis

The statistical analysis involves the use of metrics to evaluate the ontology from various dimensions. It measures the individual quality attribute of the ontology. Both the structural metrics and the schema metrics were used.

3.6.1.1 Structural Metrics

Structural measures included the total number of classes or concepts, total number of properties, total number of instances or individuals of ontology, maximum number of children, number of subclasses of the upper class in the inheritance tree in the ontology, average number of children and average number of subclass relations per class in the ontology [36]. The primary metrics of AEC3PO provide the count of the number of classes, objects, axioms, properties and individuals used in the ontology. The list of base metrics is given in Table 3. The size of ontology affects the process of ontology merging, alignment and reuse. Thus, a quantitative overview of ontology would be beneficial.

Metrics	**AEC3PO Count**
#Axioms	803
#Logical Axioms	174
#Declaration Axioms	118
#Classes	76
#Object Properties	39
#Data Properties	9
#Annotation Properties	29

Table 3 AEC3PO Metrics

3.6.1.2 Schema Metrics

The schema metrics articulated the design of the ontology. Richness, width, depth and inheritance of the developed ontology are evaluated using the schema metrics. The schema metrics of AEC3PO is given in Table 4.

Schema Metrics	Description	Formula	AEC3PO Result
Attribute Richness (AR)	Average number of attributes (slots) per class	$AR = \frac{\|ATT\|}{\|C\|}$ ATT - number of attributes for all the classes C - the number of classes	**0.63** This metric gives insight into how much knowledge about classes is in the schema. AEC3PO AR (**0.63**) indicates that each class has a decent number of attributes on the average.
Inheritance Richness (IR)	Range of distribution horizontally and vertically	$IR = \frac{\sum_{c_i \in c} \|H^C(c_1, c_i)\|}{\|C\|}$ H^c - number of inheritance relationships in class C_i C_1 and C_i – are respectively class 1 and class of order *i* from the set of classes in the ontology	**0.8** This measure describes the distribution of information across different levels of the ontology inheritance tree or the fan-out of parent classes. This is a good indication of how well knowledge is grouped into different categories and subcategories in the ontology. An ontology with a high IR would be of a horizontal nature, which means that ontology represents a wide range of general knowledge, which is the case of AEC3PO with IR = 0.8.
Relationship Richness (RR)	Number of different types of relations	$RR = \frac{\|P\|}{\|H\| + \|P\|}$ P – non-inheritance relationships	**0.69** This metric reflects the diversity of relations and placement of relations in the ontology. An ontology that contains many relations other than class-subclass relations is richer than a taxonomy with only class-subclass relationships. AEC3PO RR (~0.7) indicates that the ontology most of the relationships are other than class-subclass.
Axiom Class Ratio (ACR)	Ratio between axioms and classes	$ACR = \frac{\|Axioms\|}{\|C\|}$	**10.56** This metric is often used to assess the level of detail, expressiveness, and complexity of an ontology. It can provide insights into the comprehensiveness and depth of modelling within the ontology. A higher ratio may indicate a more fine-grained and detailed representation of concepts, while a lower ratio may suggest a more high-level and abstract ontology.

			In our case, AEC3PO ACR proves its high-level of fine granularity.
Inverse Relations Ratio (IRR)	Ratio between the inverse relations and all relations	$IRR = \frac{IOP + IFDP}{AOP + AFDP}$ IOP – InverseObject Properties IFDP – InverseFunctionalDataProperties AOP – AllObjectProperties AFDP – AllFunctionalDataProperties	**0.05** The IRR is used to measure the balance between direct and inverse relationships in an ontology. A high IRR indicates that many relationships in the ontology have corresponding inverse relationships, while a low IRR suggests that there are fewer inverse relationships relative to the total number of relationships. In our case, AEC3PO has a very low IRR. This can be enhanced in the future to improve the ontology's ability to capture bidirectional knowledge and relationships between concepts.

Table 4 AEC3PO Schema Metrics

3.6.2 Expert Evaluation

In addition to the statistical analysis, human evaluation plays a significant role in assessing the quality and usability of the AEC3PO ontology. To ensure a comprehensive assessment, we conducted a user survey involving experts from the AEC domain, researchers, and ontology engineers. This survey aimed to gather feedback on various aspects of the AEC3PO ontology. Below is a list of evaluation criteria used in the survey:

- **Ease of understanding:** Participants were asked to evaluate how easily they could comprehend the ontology, including its class hierarchy and property definitions.
- **Completeness:** Participants assessed whether the ontology covers an adequate range of concepts and properties in the AEC domain.
- **Consistency:** Participants provided feedback on the consistency of naming conventions and definitions within the ontology.
- **Usability:** The survey included questions regarding the practical usability of AEC3PO for real-world applications.
- **Extensibility:** Experts assessed the extensibility of the ontology to accommodate future changes and additions within the AEC domain.
- **Documentation:** Participants evaluated the quality and availability of documentation that accompanies the ontology

- **Overall satisfaction:** Survey participants were asked to express their overall satisfaction with the AEC3PO ontology.

- **Recommendation:** Participants indicated whether they would recommend the AEC3PO ontology to others in the same domain.

We collected valuable insights and feedback from the survey participants, which have been instrumental in refining and improving the ontology. The questionnaire used in the survey is available as an Annex, allowing stakeholders to access and review the questions that were used in the assessment (please see Annex A. The questionnaire on the AEC3PO – The Architecture, Engineering, Construction, Compliance Checking and Permitting Ontology).

The human evaluation of the AEC3PO ontology involved a diverse group of participants from various backgrounds, including researchers, PhD students, and ontology engineers. This diverse group ensured a well-rounded assessment of the ontology's quality and usability. The distribution of participants is presented in Figure 33.

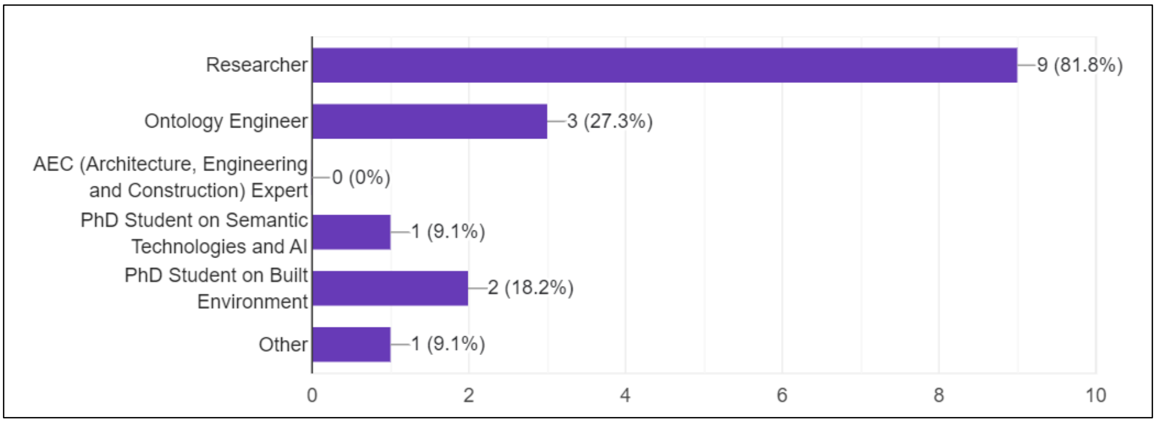


Figure 33 Distribution of participants to the AEC3PO Survey

Now, let us delve into the results for each evaluation criterion:

- **Ease of understanding:** The survey revealed that 82% of participants found the AEC3PO ontology easy to understand, as depicted in Figure 34.

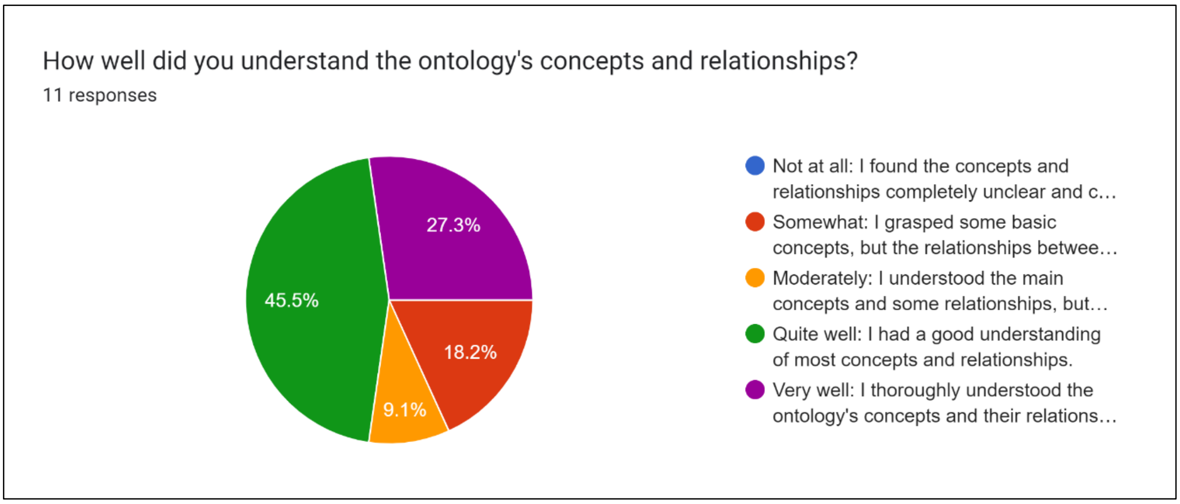


Figure 34 AEC3PO Ease of Understanding

- **Completeness:** Participants evaluated the ontology's completeness, and an impressive 100% of respondents to this question considered it to be comprehensive, as illustrated in Figure 35.

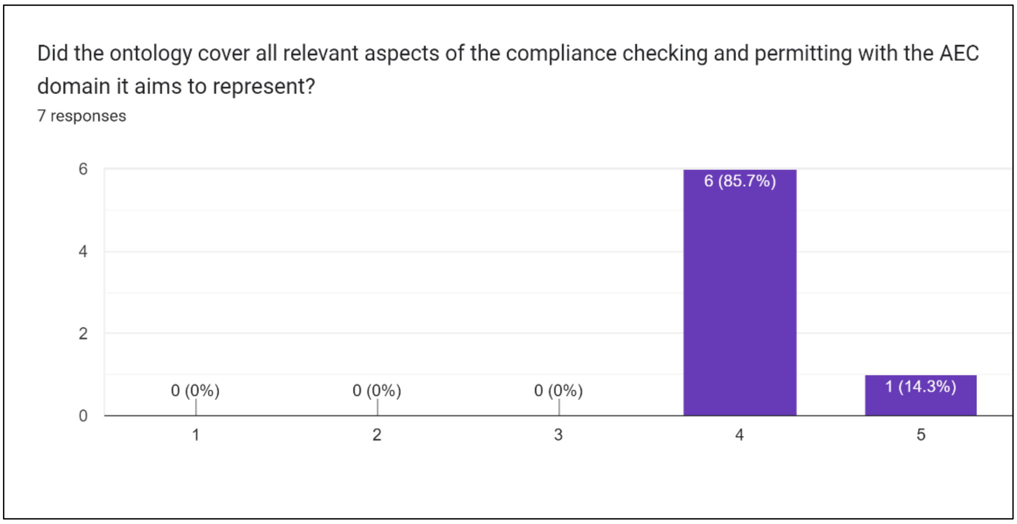


Figure 35 AEC3PO Completeness

- **Consistency:** An overwhelming 90% of participants found the ontology consistent in terms of naming conventions and definitions. The remaining 10% mentioned neighbour hearing issues. However, we do not consider it as an inconsistency. It is rather a future expansion of the ontology. The graphical representation of this result is shown in Figure 36.

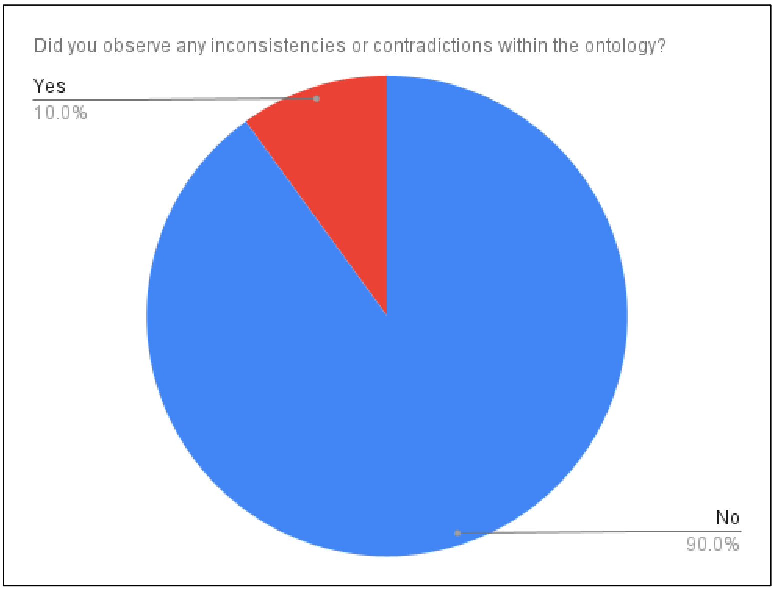


Figure 36 AEC3PO Consistency

- **Usability:** In terms of usability, 100% of respondents indicated that the AEC3PO ontology is usable. The level of usability has a different distribution among participants, but it is noteworthy that 64% of them found that the ontology is highly usable for practical applications. Figure 37 provides a visual representation of this result.

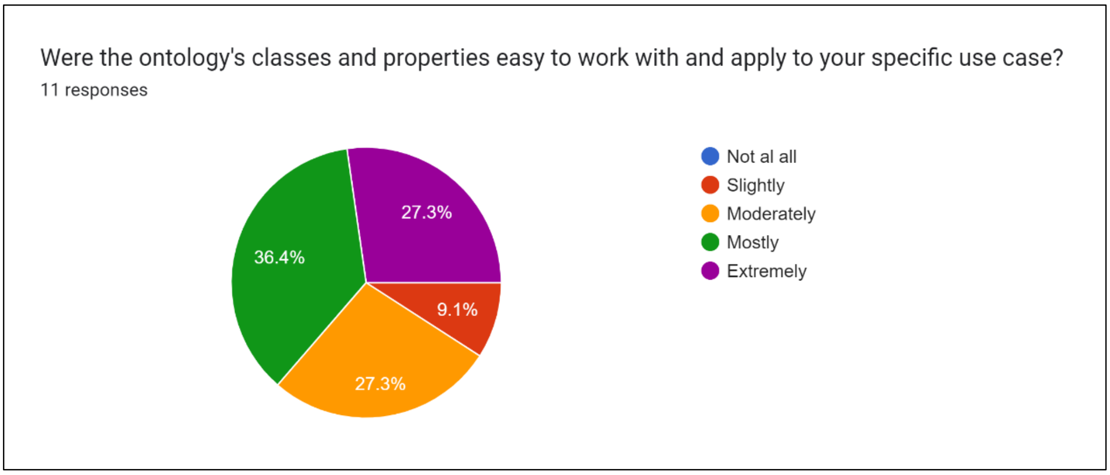


Figure 37 AEC3PO Usability

- **Extensibility:** The extensibility of the ontology was assessed, with 90% of respondents acknowledging its potential for accommodating future changes and additions within the AEC domain. Figure 38 illustrates this outcome.

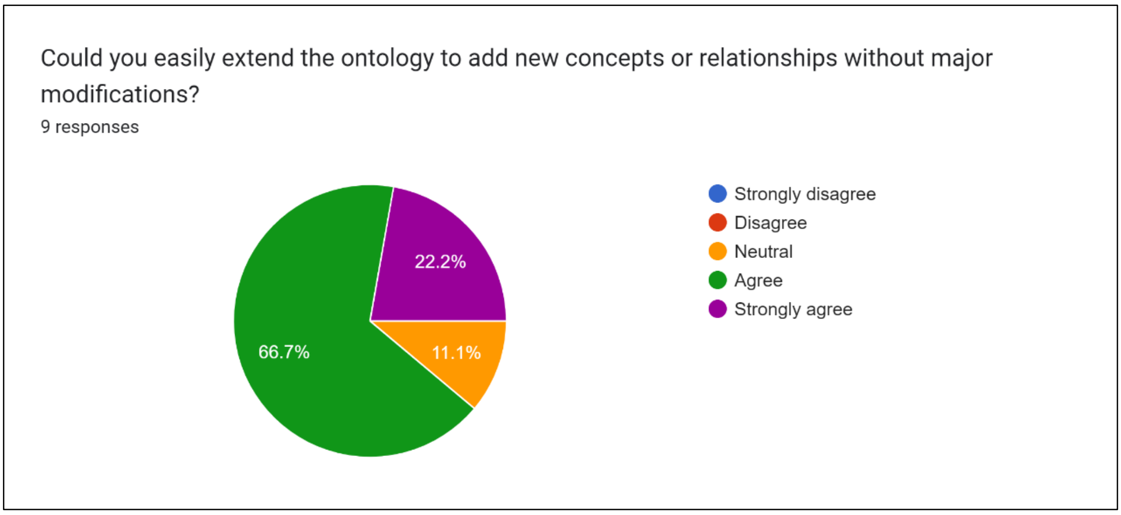


Figure 38 AEC3PO Extensibility

- **Documentation:** A significant 64% of survey participants appreciated the quality and availability of documentation provided with the ontology. The graphical representation of this result can be seen in Figure 39.

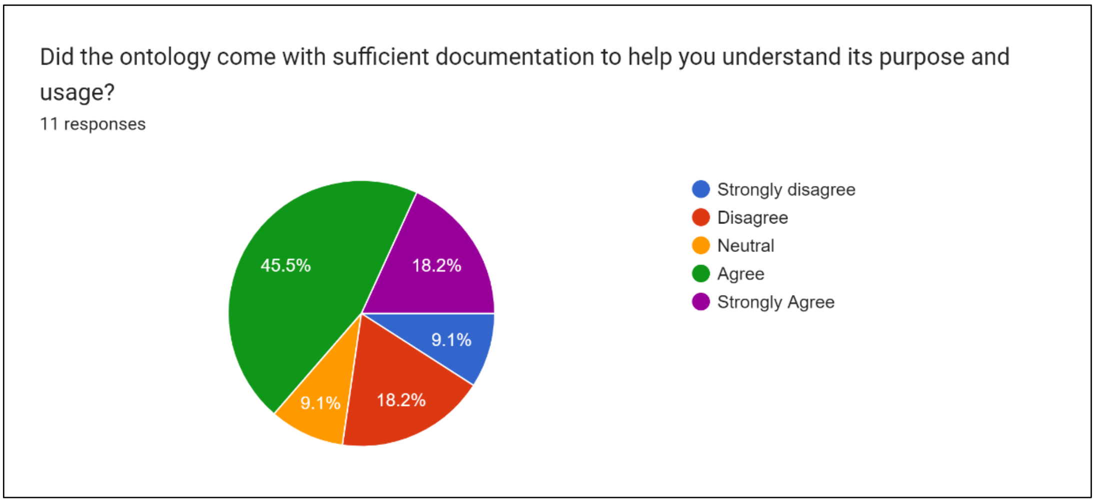


Figure 39 AEC3PO Documentation

- **Overall Satisfaction:** When asked about their overall satisfaction, 100% of respondents expressed their contentment with the AEC3PO ontology. Figure 40 visually presents this result.

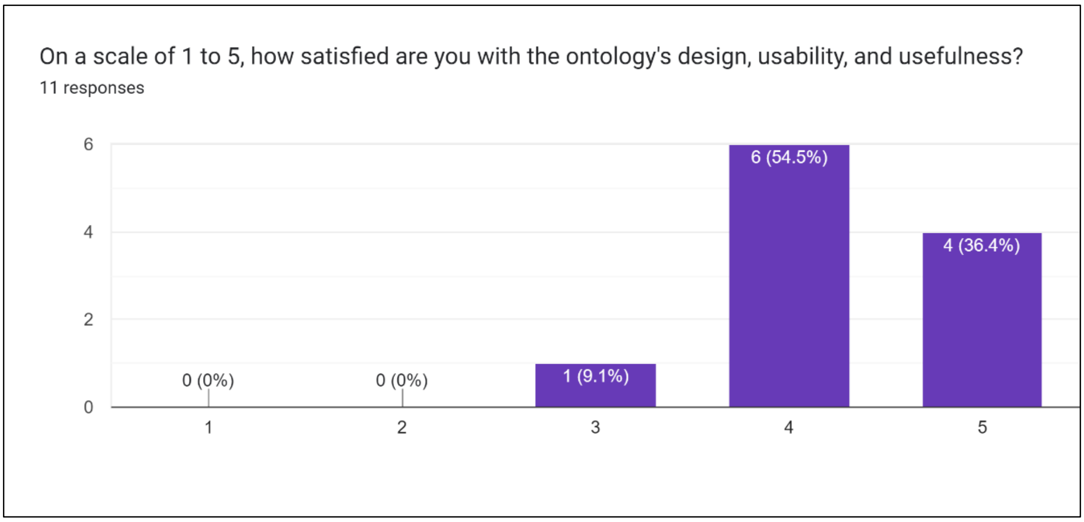


Figure 40 AEC3PO Overall Satisfaction

- **Recommendation:** The majority of participants, approximately 82%, indicated that they would recommend the AEC3PO ontology to others in the AEC domain. Figure 41 offers a graphical representation of this outcome.

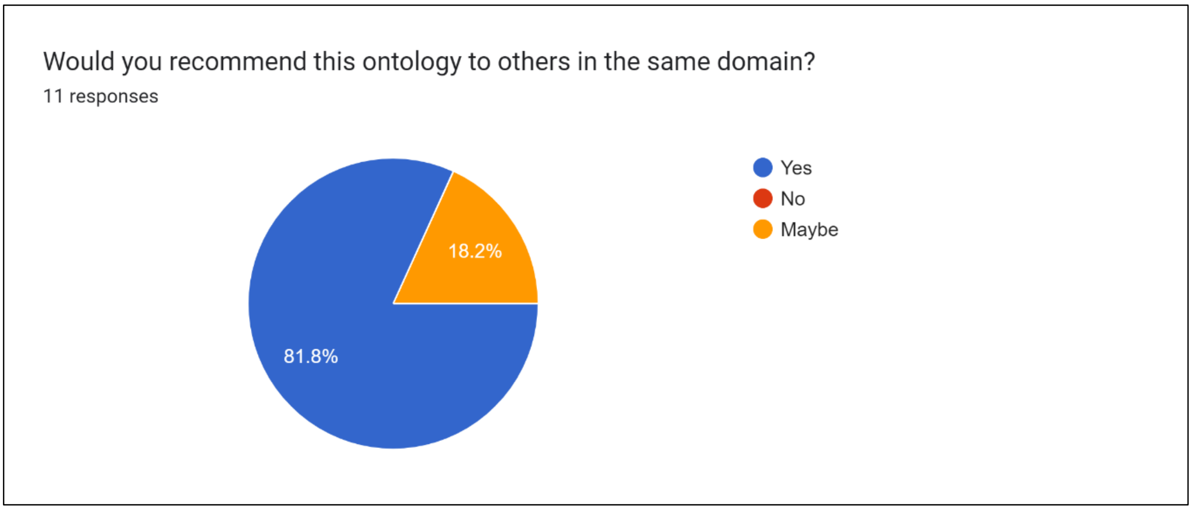


Figure 41 AEC3PO Recommendation

These survey results reflect the positive reception of the AEC3PO ontology by experts, researchers, and ontology engineers. The feedback collected from this diverse group of stakeholders has been invaluable in enhancing the ontology's quality, and it reaffirms its potential to serve as a valuable resource for the AEC community.

In addition to the questionnaire-based evaluation, AEC3PO has undergone further evaluation by being presented at various venues within the AEC domain. This exposure has allowed the AEC3PO ontology to receive valuable feedback from experts in the field. Some of the notable venues where AEC3PO has been presented include:

- The Third Building Digital Twin International Congress[5], May 3rd, 2023, Belgium.
- 11th Linked Data in Architecture and Construction Workshop (LDAC 2023), 15-16 June 2023, Italy. In this workshop, AEC3PO has been presented as part of the Industrial Talk titled *"Semantisation of Rules for Automated Compliance Checking"* [37]
- ACCORD Meetings.
- Sister Projects Meetings (Check[6] and DigiCkeck[7]).
- Journal Publication(s)

All the presentations and publications can be found in the AEC3PO GitHub public repository under the Dissemination section. The feedback and insights received during these presentations have played a crucial role in refining and enhancing the ontology, making it more robust and relevant for practical applications in the ACC domain.

3.6.3 Real-World Validation

Real-world validation involved instantiating the AEC3PO ontology with real-world scenarios from diverse use cases from demo countries presented in WP5. These use cases were carefully selected to represent different aspects of the AEC domain, ensuring that the coverage, applicability, and effectiveness of the ontology were thoroughly tested. All these use cases have been thoroughly described in Section 3.5.

It is worth noting that a total of five use cases have been employed to instantiate the AEC3PO ontology. These use cases have been selected from demo countries, demonstrating the wide applicability and coverage of the ontology.

The selection of use cases from diverse administrative areas and regulatory contexts ensures that the AEC3PO ontology can be effectively applied to a wide range of scenarios within the architecture, engineering, and construction domain. This comprehensive approach underscores the adaptability and suitability of the ontology for various compliance validation tasks and regulations across different regions.

3.6.4 Ontology Evaluation Tools

The OOPS (OntOlogy Pitfall Scanner!)[8] ontology evaluation tool is a widely used software tool for assessing the quality and performance of ontologies. It is designed to detect and identify potential issues, errors, or pitfalls within ontologies, helping ontology developers refine and improve their creations. OOPS examines various aspects of ontologies, including structural issues, logical inconsistencies, and adherence to best practices.

---

[5] https://buildingdigitaltwin.org/bdtic-2023/
[6] https://chekdbp.eu/
[7] https://digichecks.eu/
[8] https://oops.linkeddata.es/response.jsp

In the case of AEC3PO, the OOPS ontology evaluation tool was utilised to assess the different modules and components of the ontology. By subjecting AEC3PO to this evaluation, it helped identify any potential pitfalls, errors, or areas of improvement within the ontology. The results obtained from OOPS provided valuable insights into the quality of AEC3PO and its adherence to ontology development best practices.

In the evaluation of AEC3PO using the OOPS ontology evaluation tool, specific attention was given to different modules of the ontology. An example of OOPS results, which is related to the module Document, is presented below in Table 5.

Based on the findings from the OOPS evaluation, the AEC3PO ontology was refined and enhanced, addressing any identified issues or pitfalls. This iterative refinement process ensured that the ontology met the desired quality standards and was well-prepared for practical applications in the AEC domain.

Issue Type	Number of occurrences	Severity
Results for P10: Missing disjointness	1 case	Important
Results for P11: Missing domain or range in properties	1 case	Important
Results for P13: Inverse relationships not explicitly declared	6 cases	Minor
Results for P22: Using different naming conventions in the ontology	1 case	Minor
Results for P24: Using recursive definitions	1 case	Important
Results for P34: Untyped class	1 case	Important
Results for P35: Untyped property	1 case	Important
Results for P40: Namespace hijacking	1 case	Critical

Table 5 OOPS Evaluation results of the document module

3.7 Summary

The ontology section of this deliverable has provided a comprehensive insight into the AEC3PO ontology, focusing on its development, structure, alignment, applicability, and evaluation. It commenced with a thorough review of existing ontologies related to compliance checking in the built environment, establishing the context for AEC3PO. The section then proceeded to deliver an overview of AEC3PO, highlighting its modules, classes, and properties, followed by an exploration of the ACIMOV methodology employed in its development.

Additionally, the section demonstrated the real-world applicability of AEC3PO by instantiating it with real cases from demo countries, showcasing its wide coverage in the AEC domain. Finally, the section presented the results of the ontology evaluation, encompassing quantitative and qualitative analyses, ensuring its quality, usability, and effectiveness.

This holistic examination of AEC3PO confirms its potential as a valuable resource for compliance checking within the AEC domain, offering a standardised and structured approach to ensure adherence to building regulations and codes.

AEC3PO will be employed as a foundational context for the rule language discussed in the following section. It serves as the essential framework for specifying and understanding compliance rules, thereby playing a pivotal role in the development of a comprehensive rule language designed to facilitate rule-based compliance checking within the AEC industry.

4. Manual ACCORD Regulation Digitisation Methodology and Domain-Specific Language

This section will describe in more detail the manual ACCORD regulation digitisation methodology along with the representation of the results of the methodology as an instance of the AEC3PO ontology (described in Section 3). More specifically, this section will first present the requirements elicited for the digitisation process (Section 4.1). Sections 4.2-4.5 will then document each phase of the digitisation methodology in turn. Section 4.6 will then present how these rules will be eventually made available to the other components in the ACCORD semantic framework via a ruleset database. Finally, Section 4.7 will document the verification of the methodology, with Section 4.8 concluding the Section.

4.1 Digitisation Process Requirements

This section will describe the requirements that have been elicited to drive the specification of the ACCORD manual digitisation process. These requirements have either been derived from (a) previous work in ACCORD D1.2 (b) specific requirements to meet the needs of the ACCORD demonstrators, gathered through an initial analysis of the documents described in Table 1. The full set of requirements are shown in Table 6.

No	Requirement Name	Source
1	Provide an intuitive method to allow regulation experts to digitise building codes/regulations and embed rules within them without the need to write code.	User Requirements (D1.2) Requirement R5
2	Should support the integration of data dictionaries to enable mappings between regulatory terms and data schemas.	User Requirements (D1.2) Requirement R9
3	Should be able to leverage emerging Artificial Intelligence techniques, such as semantic deep learning Natural Language Processing (NLP).	User Requirements (D1.2) Requirement R10
4	The digitised format of building/codes regulations should be independent of any specific building modelling format.	User Requirements (D1.2) Requirement R13

5	Should support the use of classification systems.	User Requirements (D1.2) Requirement R14
6	Should provide the formalisation of concepts from building codes/regulations in a semantic form.	User Requirements (D1.2) Requirement R15
7	Should retain the ability for manual human input.	User Requirements (D1.2) Requirement 24
8	Provide integration with microservices.	User Requirements (D1.2) Requirement R28, R31, R34, R46, R47 and R49
9	Provide logical chaining supporting logical comparisons and appropriate comparison operators.	Demonstration Requirement
10	To be able to support standard first-order logic concepts.	Demonstration Requirement
11	To be able to promote reuse of logic between different regulatory documents.	Demonstration Requirement
12	Provide separation between scoping (filtering) and checking statements.	Demonstration Requirement
13	Support diagrams and tables.	Demonstration Requirement

Table 6 Digitisation process requirements

4.2 Context Extraction

The first stage of the manual methodology is the translation of the various non-machine-readable documents that contain the target regulations of the ACCORD project (Table 1) into a machine-readable form. To do this, a spreadsheet template was developed, and the contents of the regulations were transformed into this template. An example of this for one of the Finnish regulations is shown in Figure 42. Based on this template, a conversion tool was developed to translate this spreadsheet into a machine-readable format.

	A	B	C	D	E
1					
2	Use case:	T5.1 UC FI4	Operational safety		
3			Items to be checked in 1st sprint summer 2023		
4	Chapter	Section	Clause	Text from the regulation	Translated text in English
5	1			Yleistä	Title:General
6	1	1		Soveltamisala	Title:Scope of application
7	1	1	1	Tämä asetus koskee uutta rakennusta, rakennuksen laajennusta ja rakennuksen kerrosalaa lisäävää tilaa sekä rakennuspaikan välitöntä ympäristöä.	This Decree shall apply to new buildings, additions to buildings and spaces increasing the floor area of buildings and to the immediate vicinity of building sites.
8	1	1	2	Rakennuksen korjaus- ja muutostyössä tätä asetusta on sovellettava, jos alkuperäinen ratkaisu on turvallisuuden tai terveydellisyyden kannalta ilmeisen haitallinen. Rakennuksen korjaus- ja muutostyöt voidaan muutoin tehdä alkuperäistä ratkaisua noudattaen. Muutokset eivät saa heikentää käyttöturvallisuutta.	This Decree is to be applied to renovation and alteration work only if the original solution is evidently unsuitable for reasons related to safety or health. In other cases, renovation and alteration work can be carried out according to the original solution. Alteration work must not weaken the safety of the building.
9	1	1	3	Tätä asetusta on sovellettava rakennuksen käyttötarkoituksen muutokseen, jos rakennuksen tai sen osan käyttötarkoitus muuttuu riskillisemmäksi.	This Decree shall apply to changes to the intended purpose of a building if the intended purpose of the building or part of the building is changed such that it is associated with a higher risk.
10	2	2		Rakennuksen käyttöturvallisuus	Title:Operational safety of buildings
11	2	2	1	Pääsuunnittelijan, rakennussuunnittelijan ja erityissuunnittelijan on tehtävänsä mukaisesti huolehdittava rakennuksen suunnittelusta siten, että rakennus käyttötarkoituksensa mukaisesti täyttää käyttöturvallisuudelle asetetut olennaiset tekniset vaatimukset.	According to their tasks, the principal designer, building designer and specialised designer must ensure that the building is designed in such a way that it fulfils the essential technical, functional and architectural safety requirements for its purpose.
12	2			Putoamisen ja harhaan astumisen estäminen	Title:Prevention of falls and missteps
13	2	3		Porras	Title:Staircases
14	2	3	1	Portaan on oltava turvallinen ja tarkoitukseensa soveltuva. Portaan pinta ei saa olla liukas.	Staircases must be safe, sufficiently spacious and suitable for their purpose. The surface of stairs the must not be slippery.
15	2	3	2	Poistumisalueen sisäisen portaan vähimmäisleveys on 0,85 metriä. Tämän mitan sisäpuolelle voivat kuitenkin ulottua käsijohteet ja jalkalistat.	The minimum width of indoor stairs in an evacuation area is 0.85 metres. Handrails and skirting may, however, extend into this width.

Figure 42 Spreadsheet Format Example

This initial machine-readable format is an instance of the AEC3PO ontology utilising only the concepts from Document module (described in Section 3.3.2). This means that the instance of the ontology at this point only contains the basic structure of the document (sections, paragraphs, titles, and the text itself). An example of this presented in YAML is shown in Figure 43.


```yaml
- $id: Decree_of_the_Ministry_of_the_Environment_on_the_operational_safety_of_buildings/1
  identifier: 1
  $type: DocumentSubdivision
  title:
  - General
  hasPart:
  - $id: Decree_of_the_Ministry_of_the_Environment_on_the_operational_safety_of_buildings/1.1
    identifier: 1.1
    $type: DocumentSubdivision
    title: Scope of application
    hasPart:
    - $type: Statement
      $id: Decree_of_the_Ministry_of_the_Environment_on_the_operational_safety_of_buildings/1.1.1
      identifier: 1.1.1
      asText: "This Decree shall apply to new buildings, additions to buildings and spaces increasing the floor area of buildings and
        to the immediate vicinity of building sites."
    - $type: Statement
      $id: Decree_of_the_Ministry_of_the_Environment_on_the_operational_safety_of_buildings/1.1.2
      identifier: 1.1.2
      asText: "This Decree is to be applied to renovation and alteration work only if the original solution is evidently unsuitable
        for reasons related to safety or health. In other cases, renovation and alteration work can be carried out according to the
        original solution. Alteration work must not weaken the safety of the building."
    - $type: Statement
      $id: Decree_of_the_Ministry_of_the_Environment_on_the_operational_safety_of_buildings/1.1.3
      identifier: 1.1.3
      asText: This Decree shall apply to changes to the intended purpose of a building if the intended purpose of the building or
        part of the building is changed such that it is associated with a higher risk.
```


Figure 43 YAML Conversion Output

This initial instance of the AEC3PO ontology will provide the basis for the next stage of the methodology, the use of RASE.

4.3 RASE Application

This section describes the implementation of the RASE approach within the ACCORD digitisation methodology. It firstly provides some background on the RASE approach and then describes how the RASE meta-data will be integrated within the AEC3PO ontology instances that are being created by the ACCORD methodology.

4.3.1 RASE Background

The aim of the RASE process is to enrich the written text of the regulation with metadata that will allow other readers, including machines to pick out the key phrases and how they logically relate together.

The key element of this process is examining a document to spot the short words and phrases that can be tested and larger parts of the document that organise these phrases into a logical structure.

Whether looking for short phrases or larger elements, they will be serving one of four roles. The easiest to identify may be the *Requirements*, but there will also be *Applications*, narrowing the scope, *Selections* broadening the scope and *Exceptions* which eliminate from scope completely. This is shown in Figure 44.

The process of applying RASE to a regulation document consists of the following steps:

a. Distinguishing text that contains regulations/guidance from informative text (text that is purely descriptive or informational) and definitive text (text that defines the meaning of terms) and identifying and marking-up the sections.

b. Identifying and marking-up clauses that contain regulations.

c. Identifying and marking-up complex terms (e.g., a clause that contains sub-clauses or extra conditions), tables, etc.

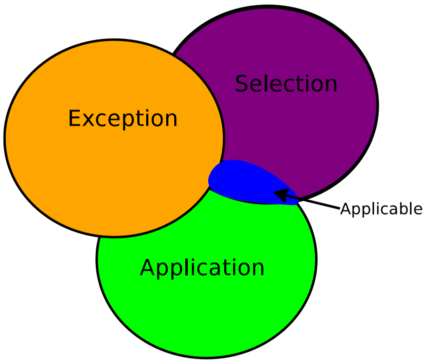


Figure 44 Determining in-scope elements.

The result of this will be a document that has a series of colours applied to it using the RASE colours. These will take the form of boxes (to identify paragraphs and clauses) and highlights to identify individual words and phrases. This creates a hierarchical structure of nested boxes, with highlighted text forming the leaf elements of this structure. An example of this is shown in Figure 45.

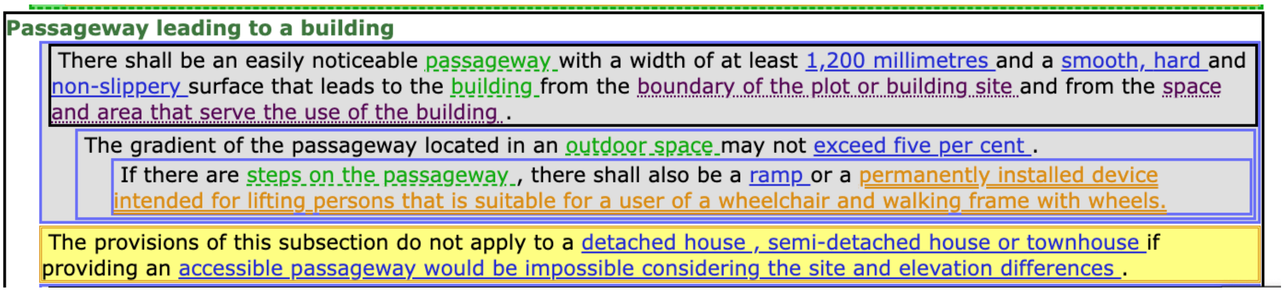


Figure 45 RASE Example

Furthermore, the colours of the highlights imply a specific logical relationship between items at the same level in the hierarchy. This enables the generation of an explicit logical representation of the document. This abstract logical representation of this is shown in Figure 46.

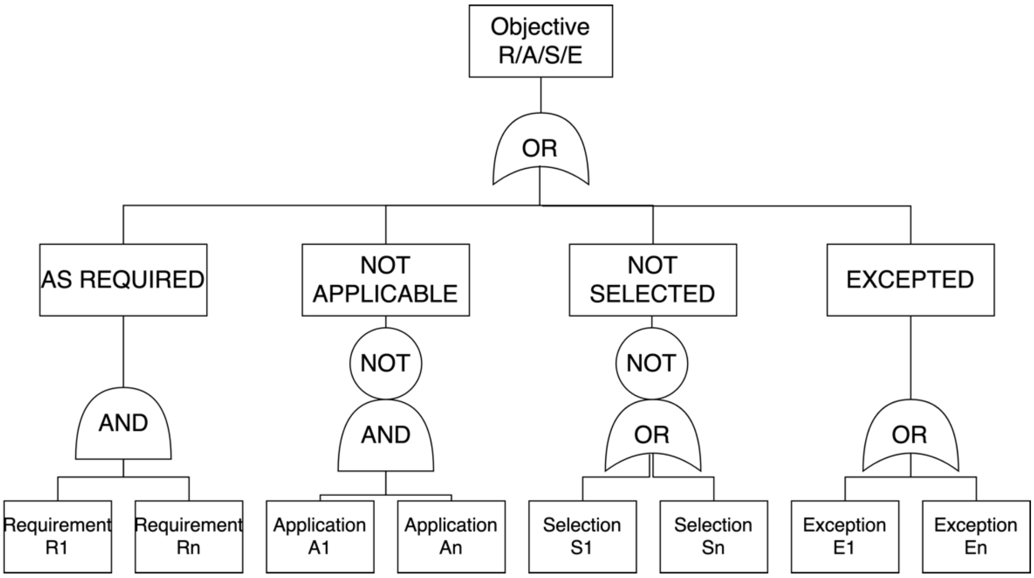


Figure 46 RASE Logical Structure (adopted from aec3.eu)

An example of this is presented in Figure 47. This figure is based Figure 45 but with annotations added to represent each box (annotations containing B) and each phrase that has been highlighted in RASE. By applying the logical formula shown in Figure 6, the logical formula for this set of clauses is:

- *Result = EB1 ∨ BR1*
- *EB1 = R1 ∧ R2*
- *BR1 = !A1 ∨ (!S1 ∧ !S2) || (R3 ∧ R4 ∧ R5 ∧ BR2)*
- *BR2 = !A2 ∨ (R6 ∧ BR3)*
- *BR3 = !A3 ∨ E1 ∨ R7*

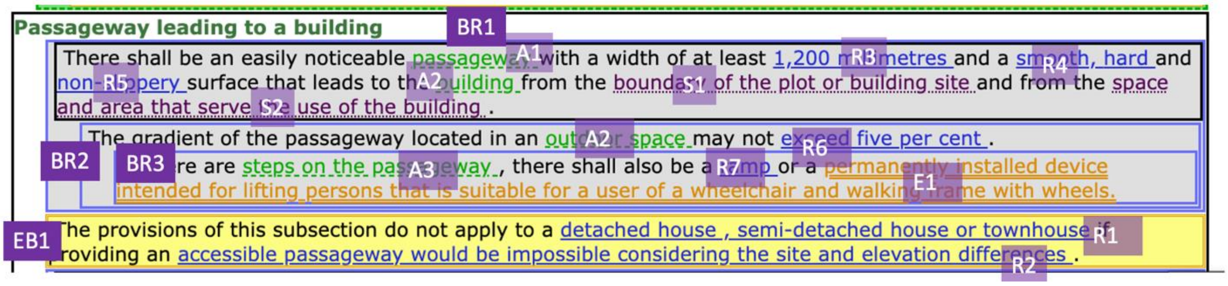


Figure 47 Annotated RASE Example

4.3.2 Representation of RASE Tags

Within an instance of the AEC3PO ontology, the RASE approach is delivered through the addition of four new classes: *Requirement, Application, Selection* and *Exception* to match the four RASE concepts. These classes can be added to any other construct within the AEC3PO ontology to indication that a particular RASE concept has been applied to this. An example of this serialisation in YAML is shown in Figure 48.

```yaml
- $id: Fire_safety_requirements_for_the_building/9
  identifier: 9
  $type: DocumentSubdivision
  title: Prevention of fire and its danger
  hasPart:
  - $type: Statement
    $id: Fire_safety_requirements_for_the_building/9.1
    identifier: 9.1
    asText: "In order to prevent fire and its risk, the possibility of fire is taken into account during the design, construction and
      use of the building, taking into account both internal and external effects, including the risk of fire and explosion caused by
      the technical systems and equipment located in the building."
  - $type:
    - Statement
    - RequirementStatement
    $id: Fire_safety_requirements_for_the_building/9.2
    identifier: 9.2
    hasInlinePart:
    - asText: Explosive
      $id: Fire_safety_requirements_for_the_building/9.2.1
      identifier: 9.2.1
      $type:
      - CheckStatement
      - ApplicationStatement
    - asText: rooms
      $id: Fire_safety_requirements_for_the_building/9.2.2
      identifier: 9.2.2
      $type:
      - CheckStatement
      - ApplicationStatement
    - $id: Fire_safety_requirements_for_the_building/9.2.3
      identifier: 9.2.3
      asText: are allowed in a building with types of use
      $type: Statement
```


Figure 48 YAML Serialisation of RASE Concepts

4.4 Expression Based Approach

The next step in the ACCORD digitisation methodology is the formalisation of the logical decision behind each RASE highlighted word/phrase. In many cases, this will be as simple as defining a single logical comparison e.g., *width > 10m*. This has two key purposes (1) it introduced a formalised term (in the case of the example width) that is related to giving the RASE highlighted word/phrase and (2) it provides an explicit formalisation of how that RASE highlighted word/phrase can be determined to be true or false.

However, there are cases where, due to the complexity of the regulations, there is need for additional complexity. Based on past experiences this includes:

- Dealing with regulations that consider connectivity and spatial considerations e.g., *a building must contain one toilet suitable for a disabled person.*

- Dealing with regulations that have phrased two separate decisions as one decision e.g., *a space must contain all applicable sanitary fittings.*
- Where regulations contain simple formulas that are useful to express explicitly within regulation document e.g., *a given building must have 1 toilet for every 10 regular occupants.*
- Combinations of the above.

Based on these use cases simple expressions can be used to meet these needs. The motivation for adopting these simple expression as opposed to utilising a more complex languages is:

- Simplicity – they can be authored by those with limited technical computing knowledge.
- Familiarity – the use of commonly recognised operators and terms will enable wide accessibility to users.
- Compatibility – the use of standard operators and structures enable expressions to be parsed and generated by a variety of software tools and user interfaces.

In order to be confident that these expressions can meet the requirements of the regulations being digitised they must meet the following requirements:

- To be able to support simple logical comparisons and appropriate comparison operators.
- To be able to support standard first order logic constructs.
- To be able to support functions to promote re-use of logic between different regulatory documents.
- To be able to support linking of the above using standard logical operators.

The syntax of the expressions is described in Figure 49, additionally the syntax has been formalised into an ANTLR grammar, to enable the rapid development of software components based on these expressions. This grammar is presented in EBNF form below:

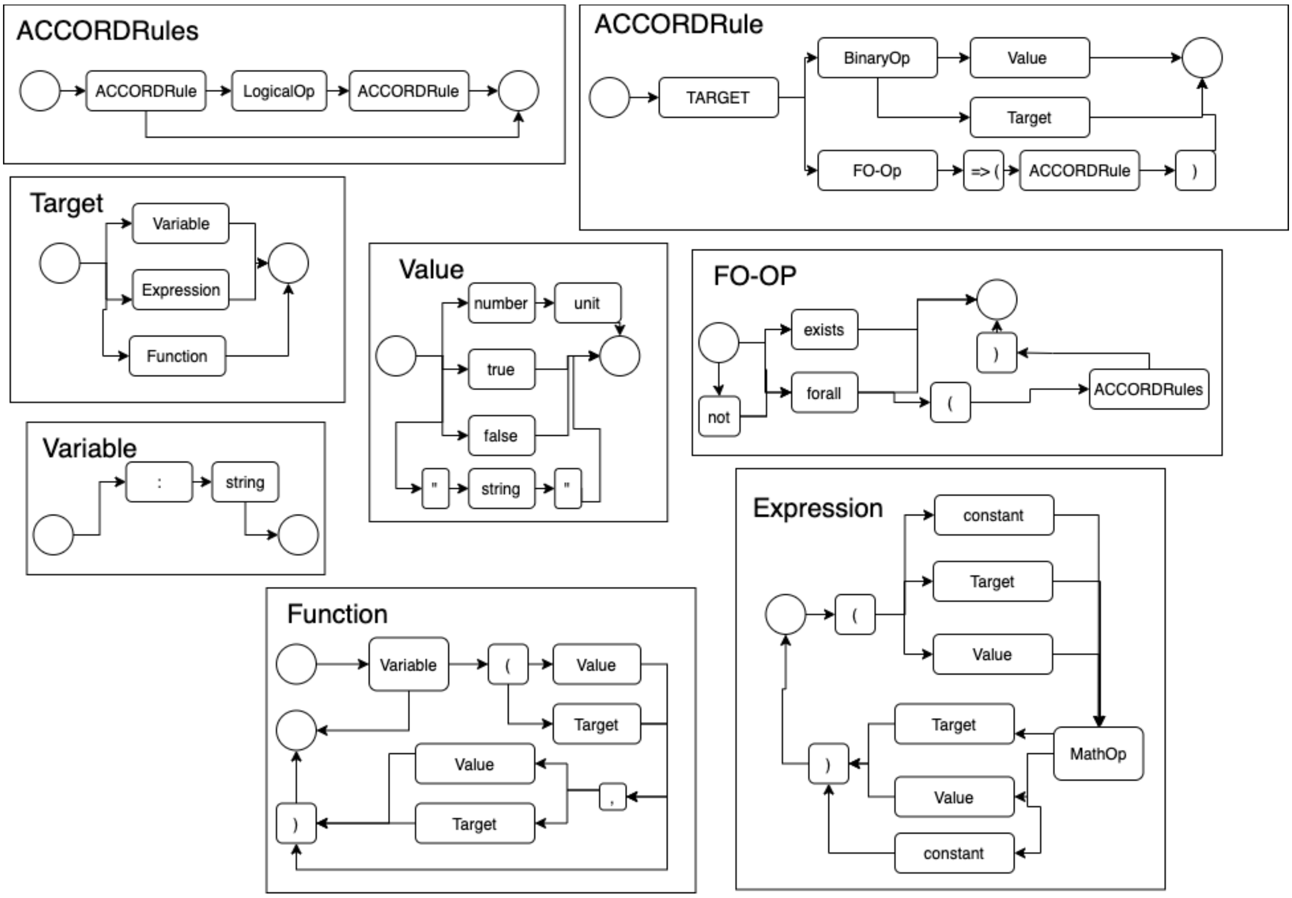


Figure 49 Expression Syntax

<AccordRules> = <AccordRule> [<LogicalOp> <AccordRules>]

<AccordRule> = <Target> (<BinaryOperator> (<Target> | <Value>) | (<FirstOrderLogic> => <OB> <AccordRules> <CB>)

<Target> = <Variable> | <Expression> | <Function>

<Function> = <Variable> <OB> [<Target> | <Value>] {, <Target> | <Value> } <CB>;*

<Expression> = <OB> (<Target> | <Number> | <Constant>) <MathOp> (<Target> | <Number> | <Constant>) <CB>

<Variable> = : <String>

<Value> = <Number> [<Unit>] | "<String>" | <Boolean>

<FirstOrderLogic> = [not] ((FORALL [<OB> <AccordRules> <CB>]) | exists)

In addition to the explicitly defined terms in the EBNF, the following terms are omitted for brevity:

- <LogicalOp> - The and/or logical operator.
- <BinaryOperator> - Any binary operator e.g., == > <=.
- <String> - A string.
- <Boolean> - true or false.
- <Number> - A number.
- <MathOp> - Any standard mathematical operator e.g. +,-,*.

- <Constant> - Any standard mathematical constant e.g., PI.
- <Unit> - A unit represented as a string.
- <OB> and <CB> - Opening and closing brackets.

Several examples of the expressions are presented in Table 7. It should be noted that each variable term (mentioned in the language in Figure 49) will be linked to a term defined in bSDD, the process of defining these terms is described in Section 4.5.

Example Expression	**Description**
:IsExternal == true	Checks if a given object is outside of the building (or part of it is outside).
:Width > 1.2 :M	Checks if the width is greater than 1.2 metres
:type == :House	Checks if the type of an object is a house.
:UsageCategory==:IV	Checks if the usage category of a object matches the classification code IV
:Walls exists => (:IsExternal == true)	Checks if a given object has a wall that is external
:AdjacentSpaces forAll => (:FireSafeDesign==true)	Checks if all adjacent spaces have a fire safe design
(:tan((:Slope *(pi/180)))*100) > 5%	Checks the following formula is > 5%. Tan(x*pi/180)*100 Where x is the slope of the given object.
:Contains exists => (:type == :LiftingDevice && :IsPermanent == true && :SuitableForWheelchair == true && :SuitableforWalkingFrame == true)	Checks if a given object contains another object that is a permanent lifting device that is suitable for wheelchairs and walking frames.

Table 7 Expression Examples

These expressions can also be formalised explicitly using the concepts within the AEC3PO ontology, specifically from the CheckMethod module. This is done by parsing the expressions into a syntax tree using the ANTLR grammar and representing the syntax tree within the ontology. An example of a syntax tree for the last example is shown in Figure 50. An example of this serialised in YAML is shown in Figure 51. It should be noted that this makes uses of the *CompositeCheckMethod*, *CategoryCheckMethod* and *BooleanCheckMethod*, check methods from the AEC3PO ontology.

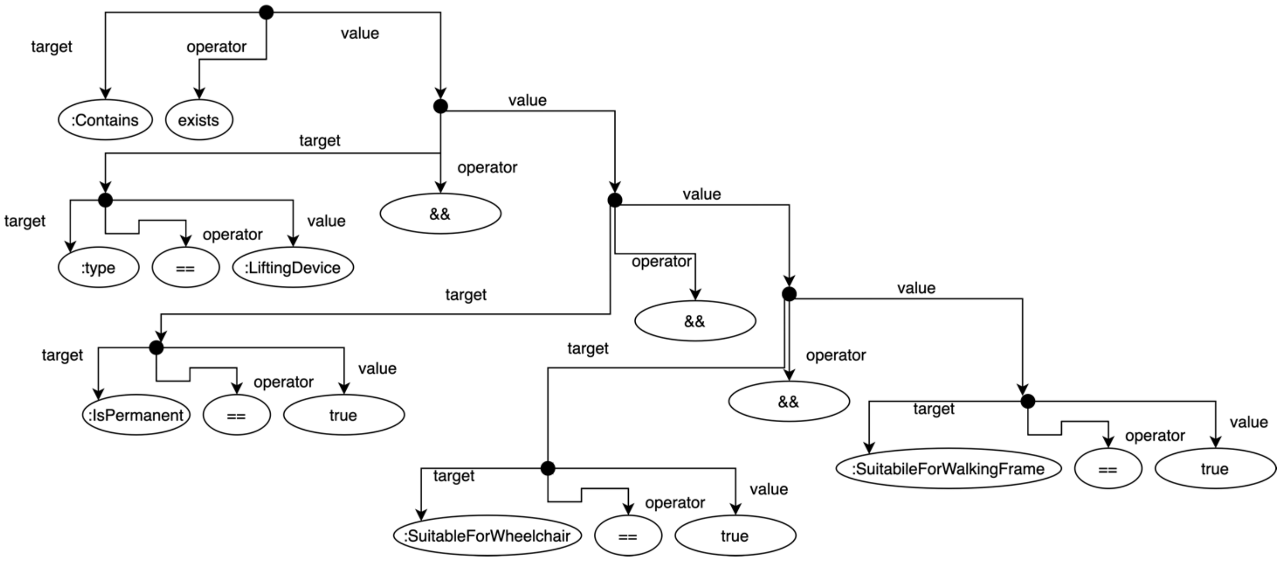


Figure 50 Expression Parse Tree


```yaml
$id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method
identifier: 2.1.2.6.6_method
hasTarget:
  $id: terms:Contains
  $type: $id
$type: CompositeCheckMethod
hasValue:
- $type: CompositeCheckMethod
  $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2
  hasComparator: CheckMethodComparator-logicalAND
  hasTarget:
    $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.1
    identifier: 2.1.2.6.6_method.2.1
    hasTarget:
      $id: terms:type
      $type: $id
    hasComparator: CheckMethodComparator-eq
    hasValue:
      $id: terms:LiftingDevice
      $type: $id
    $type: CategoryCheckMethod
  hasValue:
    $type: CompositeCheckMethod
    $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2
    hasComparator: CheckMethodComparator-logicalAND
    hasTarget:
      $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.1
      identifier: 2.1.2.6.6_method.2.2.1
      hasTarget:
        $id: terms:IsPermanent
        $type: $id
      hasComparator: CheckMethodComparator-eq
      $type: BooleanCheckMethod
      hasValue: "true"
    hasValue:
      $type: CompositeCheckMethod
      $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.2
      hasComparator: CheckMethodComparator-logicalAND
      hasTarget:
        $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.2.1
        identifier: 2.1.2.6.6_method.2.2.2.1
        hasTarget:
          $id: terms:SuitableForWheelchair
          $type: $id
        hasComparator: CheckMethodComparator-eq
        $type: BooleanCheckMethod
        hasValue: "true"
      hasValue:
        $id: Government_Decree_on_Accessibility_of_building/2.1.2.6.6_method.2.2.2.2
        identifier: 2.1.2.6.6_method.2.2.2.2
        hasTarget:
          $id: terms:SuitableforWalkingFrame
          $type: $id
        hasComparator: CheckMethodComparator-eq
        $type: BooleanCheckMethod
        hasValue: "true"
hasComparator: CheckMethodComparator-exists
```


Figure 51 YAML Expression Example

4.5 bSDD Mapping

This section will discuss the final stage in the ACCORD digitisation methodology, how each term, elicited as part of the expressions defined in the previous section, can be retrieved, or calculated. This consists of two steps: (a) identify what type of retrieval/calculation should take place, and (b) defining the specifics of the retrieval/calculation.

The remainder of this section will explore this concept, beginning with providing a description of bSDD, then the ACCORD execution hierarchy is described which provides the framework for deciding how each individual term should be retrieved/calculated. Finally, the mapping of terms to bSDD and other semantic resources is described.

4.5.1 bSDD Description

The buildingSMART Data Dictionary (bSDD) is an online service hosting classes (terms) and properties, allowed values, units, translations, relations between those and more. It provides a standardised workflow to guarantee data quality, information consistency and interoperability.

The bSDD schema is shown in Figure 52. This figure shows that bSDD has four main concepts:

- **Class:** This represents a class (type of object) within the bSDD data structure. It can have relations to one or more child classifications as well as to a set of classification properties. A classification can also be related to another classification via a classification relation.
- **ClassRelation:** This relates one classification to another and provides a type for the relationship.
- **ClassProperty:** Represents a property of a given classification, it will have a name and a datatype. It can also be related to a set of allowed/valid properties that it can take through a classification property value.
- **AllowedValue:** Represents a concrete property value, used to specify the list of possible/allowed properties for a classification property.

Use of bSDD allows the creation of custom data dictionaries as well as the use of standardised data dictionaries provided by BuildingSMART. Access to bSDD is via a web-interface as well as API access.

In practice, the construction sector uses bSDD for easy and efficient access to all kinds of standards to enrich their BIM models as well as to reference Information Delivery Specifications (IDS) and check BIM data for validity.

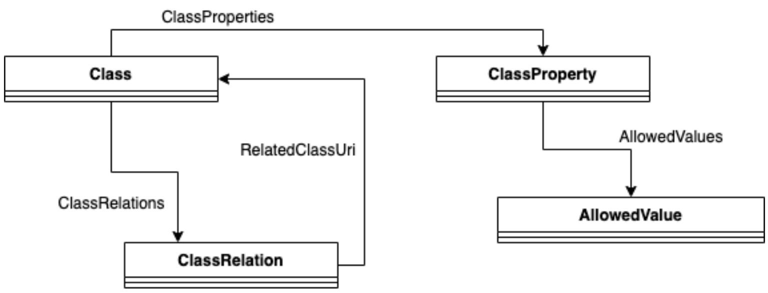


Figure 52 bSDD Schema

bSDD currently contains data dictionaries for several common use cases in the construction sector including the IFC schema and Uniclass classification codes.

4.5.2 Execution Hierarchy

The first step in providing a mapping between a term and how it is retrieved/calculated from data in a model, is to identify at a high level how it will be retrieved/calculated. To achieve this the ACCORD project has developed a hierarchy from which a specific method of execution can be selected:

1. Simple data lookup – the term is a simple data value that can be looked up from a data model.

2. Process result – the term is a result of a complex process (i.e., energy analysis) and thus must be treated as a "black-box". A specialist piece of software (termed a microservice within the ACCORD semantic framework) will need to be invoked to calculate this result.

3. Cannot be checked – no value for the term can be retrieved/calculated automatically. Thus, human input is necessary to assess this item.

If options 1 or 2 are selected, the next step is to map the term to the given data value within a data model or the specific microservice within the wider ACCORD semantic framework. This is described in the next section.

4.5.3 Mapping of terms to bSDD and other resources

To map a given term to bSDD, the following process is followed:

1. The term being considered should be categorised as either an object or a property. If a property it should be added to bSDD as a *ClassificationProperty*, if it is an object, it should be a *Classification*.
2. **If the term is an object** (e.g., Door): A *Classification* should be created in bSDD. If the term has an equivalent in any of the existing bSDD dictionaries (e.g., IFC classes, or Uniclass) then relationships between the new *Classification* and these related *Classifications* should be created.
3. **If the term is a property** (e.g., width): Firstly, the *Classification* of the property currently being considered should be inferred by the RASE meta-data present within the document (e.g., if the term is width, then what type of object the width is a property of should be inferred). This can be done automatically. Then depending on the selected execution type, one of the following should be performed:
   a. **Simple data lookup**: A relationship should be established between the inferred *Classification* and the appropriate *ClassificationProperty* in the target bSDD dictionary (depending on what model format is the target).
   b. **Process result**: A new *ClassificationProperty* should be created. The bSDD IsDynamic property is set to true and a URI for the process application is also saved.
   c. **Cannot be checked**: A new *ClassificationProperty* should be created, but no other action taken.

This will result in the creation of a series of mappings in the bSDD JSON format. This is shown in Figure 53. Specifically, this extract shows the classification "FireDoor" and its mapping to an existing *Classification* in the uniclass bSDD dictionary, as well as its relation to the IfcDoor *Classification*.

```
"Classifications": [
  {
    "Code": "FireDoor",
    "Name": "FireDoor",
    "CreatorLanguageIsoCode": "EN",
    "ClassificationRelations": [
      {
        "RelatedClassificationUri": "https://identifier.buildingsmart.org/uri/nbs/uniclass2015/1/class/TE_10_40_85_01",
        "RelationType": "HasReference",
        "Fraction": ""
      }
    ],
    "ClassificationType": "Class",
    "CountriesOfUse": "",
    "RelatedIfcEntityNamesList": [
      "IfcDoor"
    ],
    "Uid": ""
  },
```


Figure 53 bSDD Mapping

A link is also established between the term within the instance of the AEC3PO ontology and the term in bSDD. Figure 54 shows how the *Width* term within the AEC3PO ontology is formalised. The *terms:* prefix is defined on a per document basis to define the dictionary in bSDD that is being used to provide the mappings of these terms.

```
$id: Government_Decree_on_Accessibility_of_building/2.3.4.15_method
identifier: 2.3.4.15_method
hasTarget:
  $id: terms:Width
  $type: $id
hasComparator: CheckMethodComparator-ge
$type: NumericalCheckMethod
hasValue: 0.9
```

Figure 54 bSDD Mapping within AEC3PO Ontology

More specifically, a set of exemplar mappings are presented in Table 8, which shows the term identified, the execution type applied from execution hierarchy as well as the mapping itself. The mappings are drawn from the exemplar expressions in Table 8.

Term	**Execution Type**	**Mapping**
IsExternal	Simple data lookup	Pset_SpaceCommon->IsExternal IFC Property
Width	Simple data lookup	Qto_DoorBaseQuantities->Width IFC Property
Type	Simple data lookup	Special case – will map to checking against the IFC element type.
UsageCategory	Simple data lookup	Will map to the classification code placed upon the IFC object.
Walls	Simple data lookup	BoundedBy IFC Property
AdjacentSpaces	Simple data lookup	BoundedBy->RelatedBuildingElement->ProvidesBoundaries

		Will map to the spaces connected to the walls that bound this space.
FireSafeDesign	Cannot be checked – as the FireSafeDesign is a subjective judgement.	NA
Contains	Simple data lookup	ContainsElements IFC property
SuitableForWheelchair	Complex Process	Will map to a given process executing as a microservice

Table 8 bSDD Mappings

4.6 A Ruleset Database for Digitised Regulations

As described previously, once the digitisation methodology has been completed its output will be in the form of an instance of an instance of the AEC3PO ontology. In this document examples have been provided in YAML-LD, however any equivalent representation (i.e., JSON-LD, Turtle) could be used. To enable other components within the ACCORD semantic framework to make use of the digitised regulatory documents they will be hosted in a triple store enabling other components to query them via an API.

The triplestore that will be utilised is GraphDB provided by Ontotext. All instances of the AEC3PO ontology produced by the methodology documented in this section will be uploaded to the GraphDB database. How these are then used by the other components in the ACCORD semantic framework will be documented further by WP4 deliverables.

4.7 Verification

This section describes the verification of the ACCORD manual digitisation methodology. The aim of this verification is to: (1) ensure that the digitisation methodology can capture the complexity of the regulatory documents that are likely to be encountered in the demonstration cases and (2) ensure that the representation of the results of the digitisation process as an instance of the AEC3PO ontology can correctly represent both human-readable and machine-operable aspects. Subsections 4.7.1 and 4.7.2 discuss the validation of an Estonian and Finnish use case, respectively. Finally, Subsection 4.7.3 discusses the validation of the AEC3PO representation of these regulations.

4.7.1 Estonian – Fire Safety Example

To verify the digitisation methodology, example regulations from the Estonian Fire Safety regulation document have been digitised following the process. The following steps have been completed and verified:

- Conversion of the spreadsheet input into an initial AEC3PO ontology.
- Application of RASE Tags to the applicable clauses
- Authoring of expressions.
- Mapping of expressions to bSDD.

Figure 55 and Figure 56 show the results of this process showing extracts from the RASE tagged human-readable documents and YAML representation of the document.

Explosive rooms are allowed in a building with types of use I-V if they are necessary for the activities taking place in the building. Such rooms must be located near the outer wall of the building and be designed in such a way that in the event of a fire, neither the persons in the building nor the rooms adjacent to this room can be harmed. Such rooms must not be located in the basement.

If textile interior material is used in the building, it is assumed that its fire sensitivity is determined according to the relevant standard.

Flame retardant textile furnishing material must be used

1. on the evacuation route of the building;
2. in the front and intermediate curtains of the entertainment building and the sports building;
3. in a night club;
4. in the room for the provision of compulsory treatment services;
5. in a detention building;
6. in the accommodation building;
7. in the building of a welfare institution.

Figure 55 Estonian Verification - Human Readable Format


```yaml
- asText: Explosive
  $id: Fire_safety_requirements_for_the_building/9.2.1
  identifier: 9.2.1
  $type:
  - CheckStatement
  - ApplicationStatement
- asText: rooms
  $id: Fire_safety_requirements_for_the_building/9.2.2
  identifier: 9.2.2
  $type:
  - CheckStatement
  - ApplicationStatement
- $id: Fire_safety_requirements_for_the_building/9.2.3
  identifier: 9.2.3
  asText: are allowed in a building with types of use
  $type: Statement
- asText: I-V
  $id: Fire_safety_requirements_for_the_building/9.2.4
  identifier: 9.2.4
  isOperationalizedBy:
    $id: Fire_safety_requirements_for_the_building/9.2.4_method
    identifier: 9.2.4_method
    hasTarget:
      $id: terms:UsageType
      $type: $id
    hasComparator: CheckMethodComparator-eq
    hasValue:
      $id: terms:IV
      $type: $id
    $type: CategoryCheckMethod
  $type:
  - CheckStatement
  - RequirementStatement
- $id: Fire_safety_requirements_for_the_building/9.2.5
  identifier: 9.2.5
  asText: if they are
  $type: Statement
- asText: necessary for the activities taking place in the building
  $id: Fire_safety_requirements_for_the_building/9.2.6
  identifier: 9.2.6
  isOperationalizedBy:
    $id: Fire_safety_requirements_for_the_building/9.2.6_method
    identifier: 9.2.6_method
    hasTarget:
      $id: terms:NeccessaryForActivities
      $type: $id
    hasComparator: CheckMethodComparator-eq
    $type: BooleanCheckMethod
    hasValue: "true"
  $type:
  - CheckStatement
  - RequirementStatement
- $id: Fire_safety_requirements_for_the_building/9.2.7
  identifier: 9.2.7
  asText: . Such rooms must be
  $type: Statement
- asText: located near the outer wall of the building
  $id: Fire_safety_requirements_for_the_building/9.2.8
  identifier: 9.2.8
  isOperationalizedBy:
    $id: Fire_safety_requirements_for_the_building/9.2.8_method
    identifier: 9.2.8_method
    hasTarget:
      $type: FunctionCheckMethod
      $id: Fire_safety_requirements_for_the_building/9.2.8_method.1
      identifier: 9.2.8_method.1
      executes: functions:GetWalls
      hasValue: []
    $type: CompositeCheckMethod
    hasValue:
```


Figure 56 Estonian Verification - YAML

4.7.2 Finland – Accessibility Example

As a second verification, example regulations from the Finnish Accessibility regulatory document have been digitised. As with the Estonian example, the following steps have been completed and verified:

- Conversion of the spreadsheet input into an initial AEC3PO ontology.
- Application of RASE Tags to the applicable clauses
- Authoring of expressions.
- Mapping of expressions to bSDD.

Figure 57 and Figure 58 show the results of this process, showing extracts from the RASE tagged human readable documents and YAML representation of the document.

Passageway leading to a building

There shall be an easily noticeable passageway with a width of at least 1,200 millimetres and a smooth, hard and non-slippery surface that leads to the building from the boundary of the plot or building site and from the space and area that serve the use of the building .

The gradient of the passageway located in an outdoor space may not exceed five per cent .

If there are steps on the passageway , there shall also be a ramp or a permanently installed device intended for lifting persons that is suitable for a user of a wheelchair and walking frame with wheels.

The provisions of this subsection do not apply to a detached house , semi-detached house or townhouse if providing an accessible passageway would be impossible considering the site and elevation differences .

The ramp referred to in subsection 1 above shall be easily noticeable and straight with a smooth, hard and non-slippery surface, width of at least 900 millimetres and,

if the ramp is not connected to a fixed structure , a protective edge of at least 50 millimetres in height.

There shall be a horizontal landing with a length of at least 1,500 millimetres at the lower and upper end of the ramp . The gradient of the ramp may not exceed five per cent .

However, if the elevation difference is no more than 1,000 millimetres , the ramp may not have a gradient of more than eight per cent . In that case, the elevation difference of a continuous ramp may not be more than 500 millimetres , after which there shall be a horizontal intermediate landing with a length of at least 2,000 millimetres .

However, in an outdoor area the ramp may have a gradient of more than five per cent only if it can be kept in a condition comparable with that of an indoor ramp. Provisions on railings, handrails and other arrangements intended to prevent falling down and misstepping are laid down by decree issued under section 117d, subsection 2 of the Land Use and Building Act .

If parking spaces are provided for a building, an adequate number of them , but at least one , shall be intended for the use of a person with mobility and functional impairment.

Such a parking space shall have a width of at least 3,600 millimetres and a length of at least 5,000 millimetres and be marked with the International Symbol of Access.

The provisions of this subsection do not apply to a detached house, semi-detached house or townhouse.

Figure 57 Finish Verification - Human Readable Format

```yaml
identifier: 2
$type: DocumentSubdivision
title: Passageway leading to a building
hasPart:
- $type:
  - ChecklistStatement
  - RequirementStatement
  $id: Government_Decree_on_Accessibility_of_building/2.1
  identifier: 2.1
  hasPart:
  - $type: Statement
    $id: Government_Decree_on_Accessibility_of_building/2.1.1
    identifier: 2.1.1
    hasInlinePart:
    - $id: Government_Decree_on_Accessibility_of_building/2.1.1.1
      identifier: 2.1.1.1
      asText: There shall be an easily noticeable
      $type: Statement
    - asText: passageway
      $id: Government_Decree_on_Accessibility_of_building/2.1.1.2
      identifier: 2.1.1.2
      isOperationalizedBy:
        $id: Government_Decree_on_Accessibility_of_building/2.1.1.2_method
        identifier: 2.1.1.2_method
        hasTarget:
          $id: terms:type
          $type: $id
        hasComparator: CheckMethodComparator-eq
        hasValue:
          $id: terms:Passageway
          $type: $id
        $type: CategoryCheckMethod
      $type:
      - CheckStatement
      - ApplicationStatement
    - $id: Government_Decree_on_Accessibility_of_building/2.1.1.3
      identifier: 2.1.1.3
      asText: with a width of at least
      $type: Statement
    - asText: "1,200 millimetres"
      $id: Government_Decree_on_Accessibility_of_building/2.1.1.4
      identifier: 2.1.1.4
      isOperationalizedBy:
        $id: Government_Decree_on_Accessibility_of_building/2.1.1.4_method
        identifier: 2.1.1.4_method
        hasTarget:
          $id: terms:Width
          $type: $id
        hasComparator: CheckMethodComparator-gt
        $type: NumericalCheckMethod
        hasValue: 1.2
        hasUnit: :M
      $type:
      - CheckStatement
      - RequirementStatement
    - $id: Government_Decree_on_Accessibility_of_building/2.1.1.5
      identifier: 2.1.1.5
      asText: and a
      $type: Statement
    - asText: "smooth,"
      $id: Government_Decree_on_Accessibility_of_building/2.1.1.6
      identifier: 2.1.1.6
      isOperationalizedBy:
        $id: Government_Decree_on_Accessibility_of_building/2.1.1.6_method
        identifier: 2.1.1.6_method
        hasTarget:
          $id: terms:IsSmooth
          $type: $id
        hasComparator: CheckMethodComparator-eq
```


Figure 58 Finnish Verification - YAML

4.7.3 Validation of AEC3PO Instance representation

In this element of the verification exercise the regulation pipeline developed to support the ACCORD manual digitisation methodology is utilised to test the representation of the regulatory documents. Figure 59 illustrates this pipeline, which can do the following:

- Import from the spreadsheet format described in Section 4.2**Error! Reference source not found.** and produce an instance of the AEC3PO ontology in either JSON-LD, YAML-D or Turtle.
- Read an instance of the AEC3PO ontology and produce a human readable representation of that regulation document.
- Read a human readable representation of a regulatory document and create an instance of the AEC3PO ontology in either JSON-LD, YAML-D or Turtle.
- Convert between JSON-LD, YAML-D and Turtle representations of an instance of the AEC3PO ontology.

- Produce from an instance of the AEC3PO ontology a set of logical formulas (like those proposed in Section 4.3) to illustrate how the document can be evaluated.

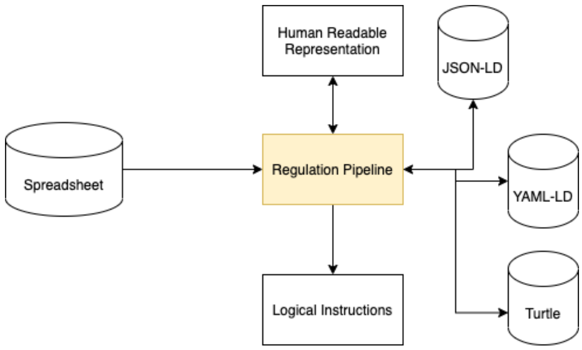


Figure 59 Regulation Pipeline Summary

Base on the use of this tool the equivalencies of all the representation formats have been validated to ensure that all conversions (except for conversion to logical instructions) are lossless and all information related to the regulatory document is retained. An example of this is shown in Figure 60.

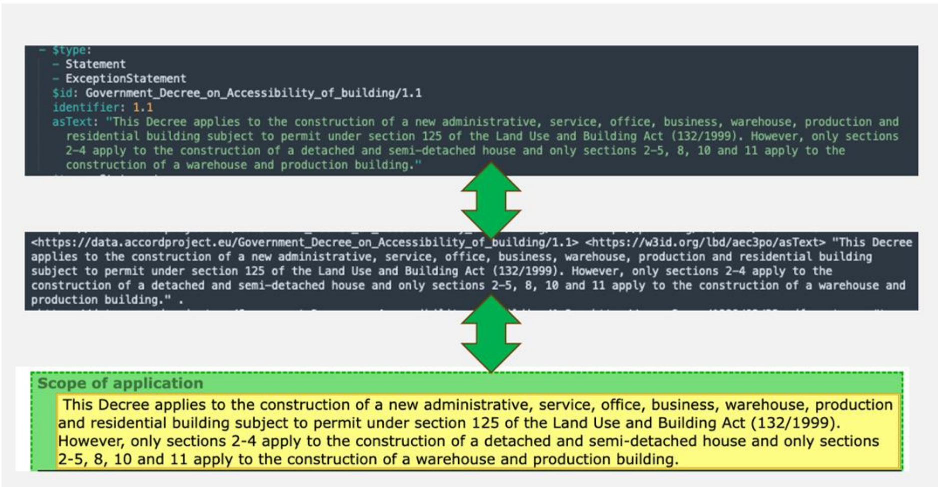


Figure 60 Equivalent Representations

This validation provides evidence that an instance of the AEC3PO ontology is fully able to retain all the information required about a given regulatory document for both human and machine consumption, regardless of the serialisation format in use.

4.8 Conclusion

This section has presented the ACCORD manual digitisation methodology. This has included descriptions of the steps involved, how the results of each of these steps are represented as instances of the AEC3PO ontology, how these ontology instances are stored in a ruleset database and how the digitisation methodology has been verified. As a final summary, at the start of this section a set of requirements for this process was elicited. Table 9 shows how each of these requirements have been met within the developed process.

No	Requirement Name	Result
1	Provide an intuitive method to allow regulation experts to digitise building codes/regulations and embed rules within them, without the need to write code.	The digitisation approach described in this section provides the means to allow regulatory experts to digitise the regulations without performing tasks such as programming. The most they would be expected to do is to author short expressions. User interfaces to support this will be developed in T2.5
2	Should support integration of data dictionaries to enable mappings between regulatory terms and data schemas.	This is met through the integration of bSDD, enabling the mapping of terms identified in a regulatory document to appropriate data schemas.
3	Should be able to leverage emerging Artificial Intelligence techniques, such as semantic deep learning Natural Language Processing (NLP).	The automated approach described will be able to leverage NLP techniques to automate the digitisation of regulations. This is documented further in D2.3
4	The digitised format of building/codes regulations should be independent of any specific building modelling format.	This is supported by the ability of the bSDD mapping element, wherein abstract terms in the regulation documents can be mapped to terms within multiple building modelling formats.
5	Should support the use of classification systems	This is supported by the ability of the bSDD mapping element, which can allow mapping of abstract terms within the regulation document to classification codes within bSDD.
6	Should provide the formalisation of concepts from building codes/regulations in a semantic form.	This is met through the ability of the digitisation methodology to utilise semantic data formats such as YAML-LD, JSON-LD and Turtle based on the formalised semantics defined in the AEC3PO ontology.
7	Should retain the ability for manual human input.	This is supported through the ability to indicate that a term within the regulatory document
8	Provide integration with microservices	This is supported through the ability to link an identified term within the regulatory documents to a specific microservice within the ACCORD semantic framework.
9	Provide logical chaining supporting logical comparisons and appropriate comparison operators.	This is achieved in two ways, through the chaining of logical concepts supported by the RASE approach, but also through the use of expression language.
10	To be able to support standard first order logic concepts.	This is achieved through the support of first order logical concepts within the expressions.
11	To be able to promote re-use of logic between different regulatory documents.	This is achieved through the support of functions within the expressions developed.
12	Provide separation between scoping (filtering) and checking statements	This is supported by the separation between scoping statements and requirements implemented by the RASE approach.
13	Support diagrams and tables	Full use of diagrams and tables is supported as per the AEC3PO ontology, they are also supported by the RASE methodology.

Table 9 Digitisation Methodology Requirement Analysis

5. ACCORD Dictionary of Terms

The ACCORD Dictionary of Terms is a specialised terminology resource that has been extracted from the regulatory documents of the five demo countries: Estonia, Finland, the UK, Spain, and Germany. This dictionary was primarily derived from the English translations of these regulations. The primary objective of the dictionary is to extract and categorise objects and properties present in the regulatory texts.

In the context of the ACCORD project, an "*object"* represents an ontological concept that signifies a specific item or element that is subject to a particular regulatory requirement. Examples of objects include "*window"* and "*fire door"*. On the other hand, a "*property"* refers to characteristics or attributes associated with these objects, such as "*width"* or "*height"*.

The development of this dictionary was influenced by the buildingSMART Data Dictionary (bSDD) approach and relies on the annotation methodology adopted in Task 2.4 of Work Package 2 (WP2). This task leverages Natural Language Processing (NLP) and Artificial Intelligence (AI) techniques to autogenerate rules from building regulations.

The annotation methodology, which will be described in detail in Deliverable D2.3, primarily focuses on entity annotation. The specific entities identified within the regulatory text are "*object"*, "*property"*, "*value"*, and "*quality"*. For the creation of the ACCORD Dictionary of Terms, the focus was exclusively on the entities "*object"* and "*property"*.

The creation process involved three key phases: data pre-processing, data annotation and data post-processing. Data pre-processing included activities such as sentence splitting and text cleaning to prepare the regulatory text for annotation and data extraction. Data annotation implemented the annotation methodology that aims to extract the entities as described above. The data post-processing consisted of removing duplicates and singularising terms to ensure consistency.

For the UK and Finland, the outcomes of Task 2.4 were utilised, which encompassed the entire regulations of both countries. From these documents, the objects and properties were extracted. In contrast, for the other demo countries, only the regulations used in Work Package 5 (WP5) were considered, aligning with the specific regulations relevant to the use cases. All objects and properties were merged and subsequently organised alphabetically to construct the entries for each dictionary of terms.

The resulting dictionary provides a valuable resource for the ACCORD project, and it is planned for these terms to be mapped to the buildingSMART Data Dictionary (bSDD) where applicable, as described in Section 4.5. Additionally, any new terms extracted will be used to enrich bSDD. Figure 61 summarises the different steps of the methodology adopted to create the ACCORD dictionary of terms.

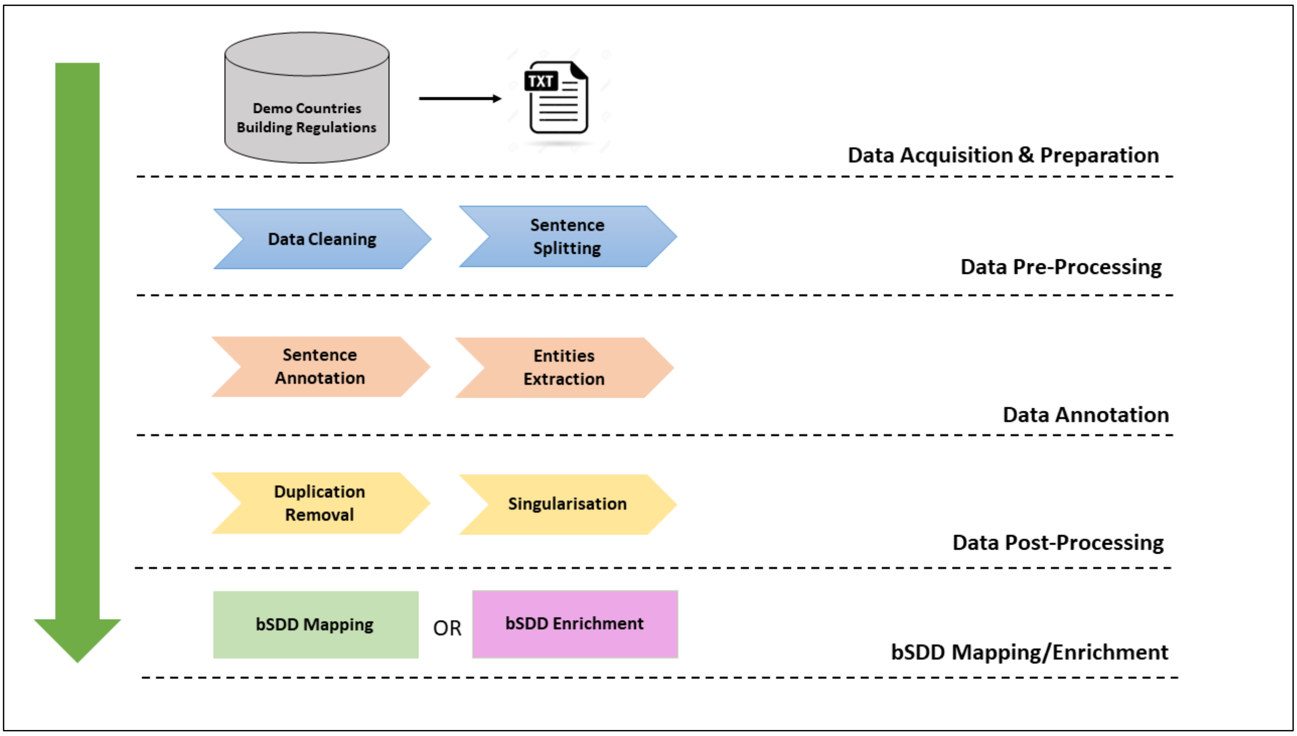


Figure 61 Steps of the development of ACCORD Dictionary of Terms

The statistics of the final dictionary are represented in Table 10, and the complete dictionary of terms is available on Github[9]. In this repository, we created separate dictionaries for each demo country, except for Finland and the UK, which share a common dictionary. This shared dictionary is a result of utilising the outcomes of T2.4 for both countries. A sample of the dictionary entries pertaining to Finland and the UK is shown in Figure 62.

Demo Country	**Number of Terms**
Finland + UK	416
Estonia	445
Germany	135
Spain	249
Total	**1245**

Table 10 Statistics of ACCORD Dictionary of Terms

[9] https://github.com/Accord-Project/aec3po/tree/main/Demo_Countries%20-%20DictionaryOfTerms

	A
1	**Terms**
2	absence
3	access
4	access point
5	access route
6	accommodation
7	accommodation premise
8	account
9	acoustic design
10	acoustic environment
11	activation
12	air
13	air barrier
14	air conditioner
15	air duct
16	air flow
17	air gap
18	air handling unit
19	air leakage rate figure
20	air pressure test

Figure 62 Sample of the ACCORD Dictionary of Terms

The ACCORD Dictionary of Terms plays a crucial role in the project's efforts to automate rule generation from building regulations and enhance the semantic understanding of regulatory texts. For instance, the dictionary of terms in the context of AEC3PO serves as a crucial bridge between the abstract ontology and the specific regulations of demo countries. It plays a vital role in concretising the instantiation of AEC3PO by linking the terminology extracted from regulatory statements to the ontology, allowing for a more nuanced and specific representation.

To facilitate this link, a new class called *ACCORDTerm* has been introduced and connected to the *AEC3PO:Statement* class as shown in Figure 63. This semantic link helps establishing a clear association between the terms found in regulatory statements and the abstract ontology, moving from a higher-level abstraction to a more concrete and context-specific representation.

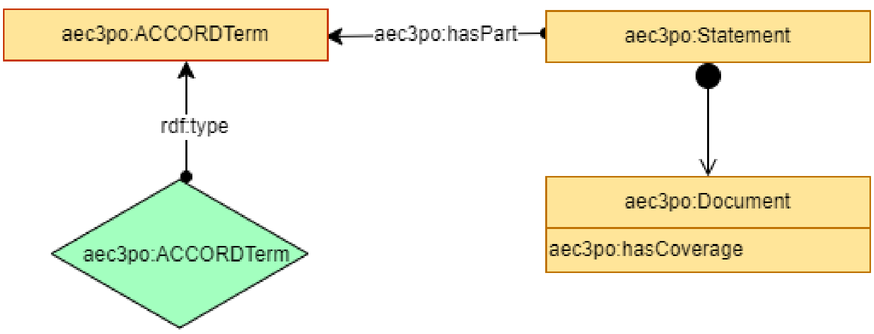


Figure 63 Linking AEC3PO to the Dictionary of Terms

For each demo country, RDF graphs have been generated, capturing the terms extracted from the text of the respective regulations. These RDF graphs are associated with a Turtle file, which is made available on GitHub at the same repository of the dictionary of terms mentioned above. These files contain the specific terms relevant to each country, forming a comprehensive dictionary of terms that aids in instantiating AEC3PO according to the distinct coverage and terminology of each demo country. This effort will be continued in WP4, where a full dictionary of terms related to ACCORD project will be created.

6. Conclusions

This deliverable has documented the outcomes of Task 2.2 Building Compliance Ontology and Task 2.3. Machine-executable Regulations. The specified objectives have been successfully realized through the accomplishment of the following goals:

- Creation of the AEC3PO ontology, which encapsulates all facets associated with building compliance and permitting within the AEC domain.
- Establishment of a methodology for the digitalisation of regulations.
- Provision of a platform-neutral format for expressing rules, utilizing semantic web technologies, and employing the ACCORD domain-specific rule language, and AEC3PO.
- Development and delivery of a graph database containing rules derived from specific regulations in the demonstration case countries.

The AEC3PO comprises components of the Compliance and Permitting Semantic Framework developed in the ACCORD project reported in Deliverable 1.2. By aligning with the objectives of the ACCORD project, AEC3PO serves as the basis of i) rule formalization methodology (Task 2.3); ii) Domain Specific Rule Language, and iii) rule formalization tool (Task 2.3), facilitating seamless communication and collaboration among experts, stakeholders, and regulatory bodies in the AEC industry.

7. References

[1] X. Tan, A. Hammad and P. Fazio, “Automated Code Compliance Checking for Building Envelope Design,” *Journal of Computing in Civil Engineering,* vol. 24, no. 2, pp. 203-2011, 2010.

[2] J. K. Lee, “Building environment rule and analysis (BERA) language and its application for evaluating building circulation and spatial program,” 2011.

[3] J. Melzner, S. Zhang, J. Teizer and H.-J. Bargstädt, “A case study on automated safety compliance checking to assist fall protection design and planning in building information models,” *Construction Management and Economics,* vol. 31, no. 6, pp. 661-674, 2013.

[4] C. Sydora and E. Stroulia, “Rule-based compliance checking and generative design for building interiors using BIM,” *Automation in Construction,* vol. 120, p. 103368, 2020.

[5] D. Moult and T. Krijnen, “Compliance checking on building models with the Gherkin language and Continuous Integration,” in *Proceedings of the EG-ICE 2020 Workshop on Intelligent Computing in Engineering*, 2020.

[6] C. Preidel and A. Borrmann, “Towards Code Compliance Checking on the basis of a Visual Programming Language,” in *ITcon*, 2016.

[7] N. Bus, F. Muhammad, B. Fiés and A. Roxin, “Semantic topological querying for compliance checking,” *eWork and eBusiness in Architecture, Engineering and Construction,* 2018.

[8] B. Zhong, L. Ding, H. Luo, Y. Zhou, Y. Hu and H. Hu, “Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking,” vol. 28, pp. 58-70, 2012.

[9] E. Hjelseth and N. N. Nisbet, “Hjelseth, Eilif and Norway Nick Nisbet. “Capturing normative constraints by use of the semantic mark-up RASE methodology.” (2011).,” 2011.

[10] T. Beach, Y. Rezgui, H. Li and T. Kasim, “A rule-based semantic approach for automated regulatory compliance in the construction sector,” *Expert Systems with Applications,* vol. 42, no. 12, pp. 5219-5231, 2015.

[11] A. L. C. Ciribini, M. Bolpagni and E. Oliveri, “An Innovative Approach to e-public Tendering Based on Model Checking,” *Procedia Economics and Finance,* vol. 21, pp. 32-39, 2015.

[12] F. Boukamp and B. Akinci, “Automated processing of construction specifications to support inspection and quality control,” *Automation in Construction,* vol. 17, no. 1, pp. 90-106, 2007.

[13] D. M. Salama and N. M. El-Gohary, “Semantic Modeling for Automated Compliance Checking,” Computing in Civil Engineering, 2011.

[14] S. Zhang, “Integrating BIM and safety: An automated rule-based checking system for safety planning and simulation,” in *Proceedings of CIB W99 conference*, 2011.

[15] S. Zhang, J. Teizer, J.-K. Lee, C. M. Eastman and M. Venugopal, “Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules,” *Automation in Construction,* vol. 29, pp. 183-195, 2013.

[16] J. Zhang and N. M. El-Gohary, “Automated Information Transformation for Automated Regulatory Compliance Checking in Construction,” *Journal of Computing in Civil Engineering,* vol. 29, no. 4, p. B4015001, 2015.

[17] S. Li, H. Cai and V. R. Kamat, “Integrating Natural Language Processing and Spatial Reasoning for Utility Compliance Checking,” *Journal of Construction Engineering and Management,* vol. 42, no. 12, p. 04016074, 2026.

[18] J. Zhang and N. M. El-Gohary, “Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques,” *Journal of Computing in Civil Engineering,* vol. 30, no. 5, p. C4016004, 2016.

[19] S. Roychoudhury, S. Sunkle, D. Kholkar and V. Kulkarni, “From Natural Language to SBVR Model Authoring Using Structured English for Compliance Checking,” in *2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*, 2017.

[20] N. O. Nawari, “A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking,” *Buildings,* vol. 9, no. 4, 2019.

[21] N. O. Nawari, “A Generalized Adaptive Framework for Automating Design Review Process: Technical Principles,” in *Advances in Informatics and Computing in Civil and Construction Engineering*, 2019.

[22] N. O. Nawari, “Generalized Adaptive Framework for Computerizing the Building Design Review Process,” *Journal of Architectural Engineering,* vol. 26, no. 1, p. 04019026, 2020.

[23] M. Messaoudi, N. O. Nawari and R. Srinivasan, “Virtual Building Permitting Framework for the State of Florida: Data Collection and Analysis,” in *Computing in Civil Engineering 2019*, 2019, pp. 328-335.

[24] Z. Zheng, Y.-C. Zhou, X.-Z. Lu and J.-R. Lin, “Knowledge-informed semantic alignment and rule interpretaion for automated compliance checking,” *Automation in Construction,* vol. 142, p. 104524, 2022.

[25] T. Mecharnia, M. Lefrançois, A. Zimmermann, E. Vakaj, A. Dridi, H. Hettiarachchi, V. Alexiev, N. Keberle, H. Tan, F. Noardo, R. Makkinga, F. Cheung, G. C. Jutglar and B. Thomas, “Existing ontologies, standards, and data domain relevant to compliance checking,” 2023.

[26] B. Zhong, C. Gan, H. Luo and X. Xing, "Ontology-based framework for building environmental monitoring and compliance checking under BIM environment," *Building and Environment,* vol. 141, pp. 127-142, 2018.

[27] L. Jiang, J. Shi and C. Wang, "Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning," *Advanced Engineering Informatics,* p. 101449, 2022.

[28] L. Morkunaite, F. Al Naber, E. Petrova and K. Svidt, "An Open Data Platform for Early-Stage Building Circularity," in *9th Linked Data in Architecture and Construction Workshop, LDAC 2021*, Luxembourg, 2021.

[29] J. Fauth and S. Seiss, "Ontology for building permit authorities (OBPA) for advanced building permit processes," *Advanced Engineering Informatics,* vol. 58, p. 102216, 2023.

[30] Y. Li, Q. Zhao, Y. Liu, X. Hei and Z. Li, "Semiautomatic Generation of Code Ontology Using ifcOWL in Compliance Checking," *Advances in Civil Engineering,* p. 8861625, 2021.

[31] X. Li, D. Yang, J. Yuan, A. Donkers and X. Liu, "BIM-enabled semantic web for automated safety checks in subway construction," *Automation in Construction,* vol. 141, p. 104454, 2022.

[32] T. R. Gruber, "A Translation Approach to Portable Ontology," *Knowledge Acquisition,* vol. 5, pp. 199-220, 1993.

[33] F.-Z. Hannou, V. Charpenay, M. Lefrancois, C. Roussey, A. Zimmermann and F. Gandon, "The ACIMOV Methodology: Agile and Continuous Integration for Modular Ontologies and Vocabularies," in *MK 2023 - 2nd Workshop on Modular Knowledge associated with FOIS 2023 - the 13th International Conference on Formal Ontology in Information Systems*, Sherbrooke, Canada, 2023.

[34] K. Clarkson, A. L. Gentile, D. Gruhl, P. Ristoski, J. Terdiman and S. Welch, "User-Centric Ontology Population," in *The Semantic Web*, Cham, 2018.

[35] S. Peroni, "A Simplified Agile Methodology for Ontology Development," in *ORE 2016: OWL: Experiences and Directions – Reasoner Evaluation*, Cham, 2016.

[36] Y. K. Kazadi and J. V. Fonou-Dombeu, "Analysis of Advanced Complexity Metrics of Biomedical Ontologies in the Bioportal Repository," *International Journal of Bioscience, Biochemistry and Bioinformatics,* vol. 7, no. 1, pp. 20-32, 2017.

[37] E. Vakaj, M. Lefrançois, A. Dridi, T. Beach, M. Gaber, G. C. Jutglar and H. Tan, "Semantisation of Rules for Automated Compliance Checking," in *LDAC2023 – 11th Linked Data in Architecture and Construction*, Matera, Italy, 2023.

Annex A. The questionnaire on the AEC3PO – The Architecture, Engineering, Construction, Compliance Checking and Permitting Ontology

AEC3PO – The Architecture, Engineering, Construction, Compliance Checking and Permitting Ontology

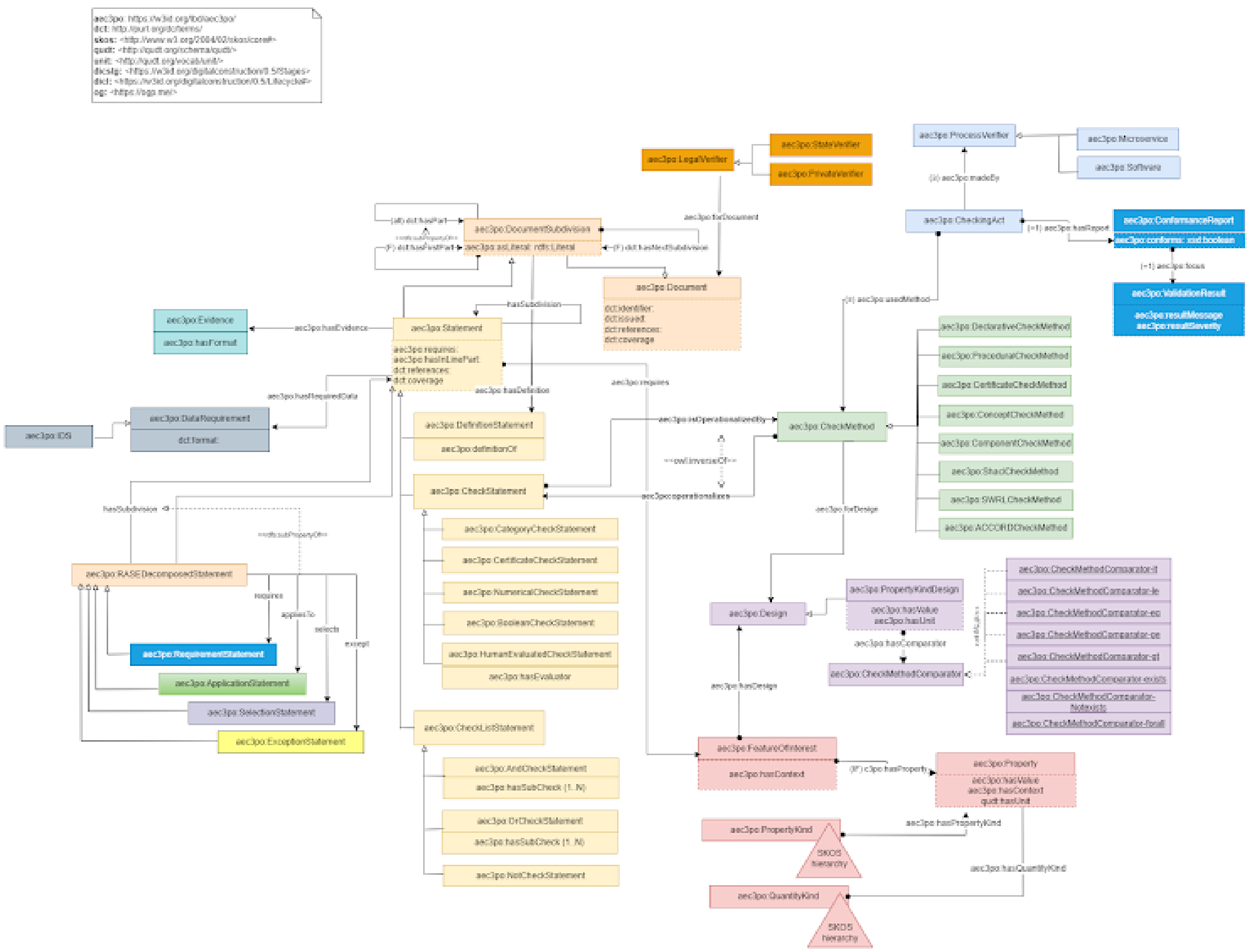
The Architecture, Engineering, Construction, Compliance Checking and Permitting Ontology (AEC3PO) is an ontology developed in the context of the Automated Compliance Checks for Construction, Renovation or Demolition Works (ACCORD) project, which is a Horizon European project that aims to digitalise permitting and compliance processes. AEC3PO is designed to represent the compliance and permitting stage in the Architecture, Engineering and Construction (AEC) domain. It models building compliance requirements, including laws, regulations, processes, and documentation. The ontology requirements are essentially derived from the rule formalisation methodology that aims to semantise regulations and provide an open format for machine-readable rules.

The ontology is built using Semantic Web technologies, adhering to standards like RDF, OWL, and SKOS. It makes use of well-known ontologies like Dublin Core Terms (DCT), Europe's Legislation Identifier (ELI), and more to create a structured and interconnected knowledge graph. This allows professionals to explore, query, and understand various aspects of the compliance and permitting processes more comprehensively.

A description of the ontology and more clear illustrations are available at the following link https://w3id.org/lbd/aec3po/

* Indicates required question

AEC3PO Overview



1. **What is your expertise?** *

Tick all that apply.

- [ ] Researcher
- [ ] Ontology Engineer
- [ ] AEC (Architecture, Engineering and Construction) Expert
- [ ] PhD Student on Semantic Technologies and AI
- [ ] PhD Student on Built Environment
- [ ] Other

Ease of Understanding

2. **How well did you understand the ontology's concepts and relationships?** *

Mark only one oval.

- Not at all: I found the concepts and relationships completely unclear and confusing.
- Somewhat: I grasped some basic concepts, but the relationships between them were unclear.
- Moderately: I understood the main concepts and some relationships, but not all.
- Quite well: I had a good understanding of most concepts and relationships.
- Very well: I thoroughly understood the ontology's concepts and their relationships.

3. **Were the classes and properties named intuitively?** *

Mark only one oval.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

4. **Have you identified any issues within the ontology? If so, could you please specify them?**

Completeness

5. **I do possess the necessary expertise or knowledge to address this section.** *

Mark only one oval.

- ◯ No - please move to the next section.
- ◯ Yes - please complete the questions under this section.

6. **Did the ontology cover all relevant aspects of the compliance checking and permitting with the AEC domain it aims to represent?**

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Completely

7. **Were there any key concepts or relationships that you felt were missing?**

Mark only one oval.

- ◯ No
- ◯ Yes

8. **If you answered Yes to the previous question, please provide the key concepts and relationships that you think they were missing.**

____
____
____
____
____

Consistency

9. **Did you observe any inconsistencies or contradictions within the ontology?** *

Mark only one oval.

- No
- Yes

10. **If you answered Yes to the previous question, please specify the inconsistencies.**

11. **Were the constraints and axioms defined in the ontology coherent and accurate?** *

Mark only one oval.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

Usability

12. **Were the ontology's classes and properties easy to work with and apply to your specific use case?** *

Mark only one oval.

- Not al all
- Slightly
- Moderately
- Mostly
- Extremely

13. **Did the ontology offer relevant and well-defined documentation for each concept?** *

Mark only one oval.

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

14. **Have you identified any issues within the ontology usability? If so, could you please specify them?**

Extensibility

15. **I do possess the necessary expertise or knowledge to address this section.** *

Mark only one oval.

- No - please move to the next section.
- Yes - please complete the questions under this section.

16. **Could you easily extend the ontology to add new concepts or relationships without major modifications?**

Mark only one oval.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

17. **Did the ontology's design support future updates and additions?**

Mark only one oval.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

18. **Have you identified any issues for the ontology extensibility? If so, could you please specify them?**

____
____
____
____
____

Documentation

19. **Did the ontology come with sufficient documentation to help you understand its purpose and usage?** *

Mark only one oval.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

20. **Were examples provided to illustrate how to use the ontology effectively?**

Mark only one oval.

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

21. **Have you identified any issues within the documentation? If so, could you please specify them?**

____

____

____

____

____

Overall Satisfaction

22. **On a scale of 1 to 5, how satisfied are you with the ontology's design, usability, and usefulness?** *

Mark only one oval.

	1	2	3	4	5	
Very	◯	◯	◯	◯	◯	Very satisfied

23. **Would you recommend this ontology to others in the same domain?** *

Mark only one oval.

◯ Yes

◯ No

◯ Maybe