Multi-requirements ontology engineering for automated processing of document-based building codes to linked building data properties

S Zentgraf[1], P Hagedorn[1] and M König[1]

[1] Chair of Computing in Engineering, Ruhr University Bochum, Bochum, Germany

E-mail: sven.zentgraf@ruhr-uni-bochum.de

Abstract. The lifecycle of a building is characterized by precise compliance with regulatory specifications for both traditional planning as well as digital planning methods such as Building Information Modeling (BIM). Currently, this regulatory information is mostly available in the unstructured and non-machine-readable form of guidelines, regulations, and standards. The acquisition and conversion of unstructured data into structured semantic knowledge bases for the use in the BIM processes are increasingly being automated using natural language processing in ongoing research. Since ontologies provide such techniques that allow raw data to be formally transformed into domain knowledge, in this paper an ontology is developed based on the data format for data catalogs, properties, and groups of properties described in the ISO 23386. The contribution of this paper is a systematic requirements analysis based on existing literature, standardization, and existing ontologies and the implementation of an ontology accordingly. Applied to a feature of interest, an exemplary property validation using the SPARQL Protocol and RDF Query Language (SPARQL) and generated Shapes and Constraint Language (SHACL) shapes is performed to show how the collected data can be used for automatic constraint checking.

1. Introduction

Building Information Modeling (BIM) enables comprehensive constraint checking on the basis of the data stored in the building models. Moreover, linked data and knowledge graphs are being increasingly utilized in Architecture, Engineering, Construction, and Operation (AECO) to overcome file-based data transfer issues, optimize data availability, and assist an open data exchange [1]. Oraskari et al. [2] showed that Linked Building Data (LBD) and SHACL allow automatic checking procedures on building models converted to LBD ontologies. However, these building codes are often only available in unstructured, non-machine-readable documents written in natural language [3].

This paper aims to assist the entity recognition in natural language building codes proposing a comprehensive data structure to cover the relations between terms in the building codes, providing the value constraints of the terms, and capturing additional metadata. Therefore, this paper outlines the requirements for an ontology representing structured knowledge from building codes based on the analysis of: (1) requirements from entity recognition literature, (2) requirements from standardization, and (3) requirements for the usage with established ontologies.

The research on Natural Language Processing (NLP) approaches for entity recognition in AECO are multifaceted [4]. Thus, a short literature review on the latest approaches highlights the common data structure needed to represent the results of NLP algorithms on building codes. Thereafter, the literature section looks at specific aspects covered by [5], which provides the fundamental concept for the development of the ontology.

General requirements for developing ontologies for the AECO domain are considered in this paper as discussed by Costin and Eastman [6]. According to their study, clear and concise terminology, consistency, reusability, extendibility, and reliability are the main requirements for an ontology in the AECO domain. Moreover, the requirements from the domain must be fulfilled sufficiently by the ontology. Regarding this, ontology requirements engineering is an elementary method for considering both the non-functional requirements mentioned above and the functional requirements from the application domains that are identified in this paper.

It is considered how the approach in this research can be integrated into existing ontology-based approaches like the Building Topology Ontology (BOT) [7] or the Ontology for Property Management (OPM) [8]. Moreover, a demonstration shows the feasibility of this approach providing queries using SPARQL and SHACL constraint shapes on example data.

2. Literature Review

This section emphasizes the previous research carried out on the three sources used in the subsequent requirements analysis. First, related literature on current entity recognition approaches for building code interpretation is introduced in Section 2.1. Section 2.2 highlights the current efforts to maintain properties using the models and processes specified in ISO 23386. Section 2.3 denotes the state of the art of ontologies and patterns for representing properties and their usage in common research.

2.1. Entity recognition approaches

The checking of specifications from building codes, regulations, and standards plays an important role in the construction industry. The stakeholders of a construction project are subject to precise guidelines in design and realization, which must be consistently proven. For Automatic Compliance Checking (ACC) of building designs, natural language specifications in regulatory documents need to be converted into machine-readable constraints [9, 10]. The conversion of these specifications is currently done by experts and is therefore associated with a high workload and error-prone [3]. In the following, research is presented that deals with the extraction of regulatory information using Named Entity Recognition (NER). NER is a sub-discipline of natural language processing (NLP) and aims at extracting entities from a text and assigning them a label from a set of predefined classes [3].

In a systematic literature review, Fuchs et al. [9] examined 41 research papers on the topic of automatic building code computerization published since 2000. They compared the techniques used, the results, and the limitations of the approaches studied. The result of their study was that the examined approaches generally worked well. Nevertheless, the implementations had low scalability and were highly dependent on the quality of the defined rules and the knowledge base.

Schönfelder et al. [3] proposed a Deep-Learning-based NER approach. They used a supervised Deep-Learning transformer model based on a pre-trained GBERT Model to extract relevant terms from corpora of German regulatory documents. The algorithm needs no user interaction and achieves a performance score of over 95% precision and 95% recall, given a set of 12 classes. The authors point out that determining how much the presented approach outperformed the rule-based procedures is still to be tested [3].

Ultimately, they state that the extracted information needs to be processed and converted into BIM data format to make it usable. To transform this unstructured data into a semantic construct, the work in this paper is important to provide a common standardized data schema for extracted information for further use in the BIM lifecycle.

2.2. Maintaining properties according to ISO 23386

In the AECO industry data catalogs form a structure for the unambiguous classification and differentiation of objects and serve as a basis for the information exchange. In practice, this exchange of information across domains is only possible to a limited extent due to the specialized terminology and non-unified properties. In particular, for the continuous use of data throughout the entire building lifecycle, information from building codes, regulatory documents, planning data, and manufacturer information can be linked together to the digital representation of a building element in the open IFC standard [11].

The ISO 23386 includes general instructions for the description, creation, and maintenance of properties in interconnected data catalogs to ensure a quality assured, seamless exchange of information between stakeholders in the BIM lifecycle. The implementation of the standard and the definitions of comprised management rules are intended to establish interoperability between data catalogs and digital tools [5]. Alani et al. [11] address this problem for water infrastructure projects by proposing awork based on Semantic Web technologies for transformation of Product Data Templates (PDTs) into ontologies. The authors recommend manufacturers or suppliers to provide their product data in an RDF-based format regulated by the ISO 23386. With the buildingSMART Data Dictionary (bsDD) [12], buildingSMART also offers an online service for maintaining classifications and properties. With this services, standardized workflows can be established to ensure data quality and interoperability within the building lifecycle. In the upcoming publication, the conformity of the bsDD data to ISO 23386 will be a new feature [12].

2.3. Ontologies and patterns for representing properties

The modeling of properties for any features of interest (FoI) of digital building models in Resource Description Framework (RDF)-based datasets is well researched [8]. The introduction of ifcOWL can be seen as the starting point for accessing building element properties in a Semantic Web context [13]. Using the ifcOWL ontology, the property sets of IFC STEP files are converted to ifcOWL equivalents and the different property types and their values are transferred into resources and literals. Generally, the W3C Linked Building Data Community Group defines and maintains the core ontology BOT to represent the topology of buildings and structures and to allow domain-specific ontologies to attach to this core data model [7]. Investigations into the compatibility of the BOT with domain ontologies and ontologies that specifically define properties for building entities were carried out by Schneider [14].

Bonduel et al. [15] examined the conversion from IFC STEP to several construction ontologies. Moreover, Bonduel et al. [15] and Rasmussen et al. [8] referred to the definition of levels of complexity (L1 to L3) for property assignments to FoI. The classification of these properties is based on the relation of the FoI and the property value. The complexity is defined depending on the number of steps needed to navigate from one property to another via the RDF graph. On this basis, Rasmussen et al. [8] developed the OPM. This ontology enables users to define properties, manage the asserted or calculated value, and the define provenance and the status (e.g., asserted or deleted) of the value. The OPM ontology is, e.g., used in the Digital Construction Ontologies (DICO) suite provided by Zheng et al.[16] for asserting properties for building entities. Moreover,the Building Product Ontology (BPO) [17] is considered for modeling and attaching attributes (i.e., properties) to a building product.

3. Research Methodology

The methodology of this paper is mainly focusing on ontology requirements analysis and, as a result, on the ontology development and demonstration. For ontology engineering, as for general software engineering tasks, requirements need to be defined to ensure the expected quality and the suitability for the application case. Therefore, methodologies for ontology engineering employ requirements specification [18, 19]. Poveda-Villalón et al. [20] provide general requirements for developing ontologies according to the FAIR (Findable, Accessible, Interoperable, Reusable) data principles and best practices for applying these to developed vocabulary. Ontology requirements can be specified in the Ontology Requirements Specification Document (ORSD) [18]. Moreover, the Linked Open Terms (LOT) methodology employs the requirements engineering according to the ORSD as well [19]. The ORSD defines a set of templates and tasks for the effective specification of functional and non-functional requirements and the intended users and use cases. Thus, the requirements analysis for the ontology development can be carried out systematically, serving as a basis for the further development steps of ontology implementation. The functional requirements, denoted as competency questions in natural language, support the development process in identifying the main elements and their relationships [21].

In this research, for the requirements analysis the procedure defined by Poveda-Villalón et al. [19] as the LOT methodology is used. The requirements formalization is done in a template of the ORSD [18]. After requirements have been specified and formalized, the implementation of the ontology is conducted by performing the conceptualization, reusing ontologies and patterns, encoding the ontology using the Web Ontology Language (OWL) [22], and evaluating the ontology based on the identified requirements. The resulting ontology is presented and demonstrated using SPARQL queries and SHACL shapes on an ABox data set modeled according to the ontology.

4. Requirements analysis

The requirements analysis is conducted in seven steps defined by the LOT methodology and formalized with the ORSD presented in Table 1 as defined in the methodology. The definition of the purpose and the scope of the proposed ontology can directly be taken from the Table 1 (steps 1 and 2, respectively). Other steps of the requirement analysis require more explanation and are outlined subsequently.

4.1. Purpose and scope

The ontology should be the structural basis for transferring unstructured guidelines or building codes into a structured class hierarchy using NLP methods. It should provide a data schema to represent the ISO 23386 data model and should enable the effective usage and validation of properties and property groups (summarized as information elements) throughout the lifecycle of a building. The purpose of the ontology is the identification, description, creation, and maintenance of properties and group of properties in interconnected data catalogs to ensure a quality-assured, seamless exchange of information between participants in BIM processes. Thus, it should simplify the transfer and exchange via data catalogs and allow linking properties to building models via linked data approaches. It should provide a data structure for developing repositories using standardized definitions of building elements and properties.

4.2. Intended users and use cases

For the development of the ontology, four use cases and intended end-users are identified (see Table 1 and Figure 1). First, the ontology could be used by **guideline bodies** and **standardization institutions** to create a digital representation of terminology provided in guidelines or standards. Therefore, NLP algorithms were examined for extracting information elements as well as their provenance information, boundary values and example data. Having information elements described in the ontology, these could be utilized for defining the Level Of Information Need (LOIN) for building models by the **clients**, assigning the required properties to building models by the **BIM modeler** and verifying the asserted information as a **BIM**

manager based on the LOIN and checking templates created from the properties. These templates could, for instance, be Model View Definitions (MVD) or SHACL shapes [2].

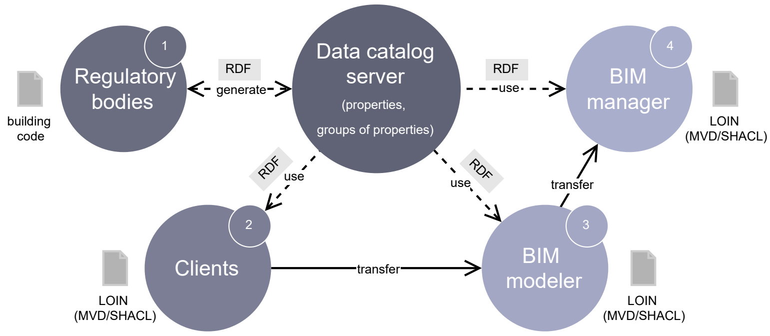

Figure 1: Use case definition, intended actors, and data exchange

4.3. Data exchange identification

The use cases and intended end users are the basis for the identification of data exchanges. The data catalog server in Figure 1 could be a central repository providing information elements via RDF-based formats. The regulatory bodies create data and share it on a data catalog server. Provenance and metadata is provided during the generation of the data. Access rights to the data and the technical implementation of the server is not relevant at this point of the requirement analysis. Clients, BIM modelers and BIM managers can retrieve data from the data catalog server in RDF-based formats. The exchange of LOIN configured through properties between Clients, BIM modelers and BIM managers is not further specified, but can be processed by means of properties in RDF format.

4.4. Functional requirements

The functional requirements are derived into three groups of competency questions (c.f. Table 1, CQG1 to CQG3) from the three sources of related work on building code entity recognition (CQG1), standardization of ISO 23386 (CQG2), and existing ontologies (CQG3).

4.4.1. CQG1. Requirements from entity recognition approaches During the analysis of the functional requirement based on the NLP algorithms, five competency questions (CQ1.1-CQ1.5) were identified. After the creation of the label classes for the NER, it must be clarified how and whether the assigned labels can be mapped to information elements (CQ1.1). Similar to the work of Schönfelder et al. [3], classes for the NER must be created for the considered use case.

For properties representing physical characteristics, it must be pointed out how boundary values can be extracted from numerical values (CQ1.2). For value pairs representing boundary values, the data model of the standard provides an attribute in which boundary value pairs can be stored in combination with a physical unit.

The provenance information of an information element is indispensable for tracing the origin of the specification at any time. Therefore, a set of information needs to be formulated which denotes all relevant provenance data (CQ1.3). Similar to CQ1.1, one or more classes must be created within the NER for this set.

While CQ1.2 deals with the mapping of boundaries for primitive data types, CQ1.4 and CQ1.5 handle the question which non-primitive data types can be mapped within a property. The less complex non-primitive data type is an enumerated type, which consists of a fixed set

Table 1: Ontology requirements specification (template from `https://github.com/oeg-upm/LOT-resources`)

Ontology Requirements Specification Document		
1. Purpose		
- structural basis for transferring unstructured guidelines or building codes into a structured class hierarchy - data schema to represent a ISO 23386 data model - effective assignment, usage and validation of properties and property - simplified transfer and exchange via data catalogs - linking properties to building models via linked data approaches		
2. Scope		
- the description, creation and maintenance of properties and group of properties in interconnected data catalogs - quality-assured, seamless exchange of information between participants in BIM processes - provision of central repositories using standardized definitions of building elements and properties		
3. Implementation Language		
Web Ontology Language (OWL)		
4. Intended End-Users		
User 1. Guideline bodies that want to digitize their guidelines User 2. Clients that want to define their information need User 3. BIM modelers that want to assign properties User 4. BIM managers that want to define rule checking sets		
5. Intended Uses		
Use case 1. Used as a data model for NLP procedures extracting properties and property groups Use case 2. Used as a queryable database for properties and property groups for use in LOINs Use case 3. Used for assigning properties and values to building elements Use case 4. Used for integrating properties into rule checking templates like MVDs		
6. Ontology Requirements		
a. Non-Functional Requirements		
NFR 1. Concise terminology (c.f. [6]) NFR 2. Consistency (c.f. [6]) NFR 3. Extendibility (c.f. [6]) NFR 4. Reliability (c.f. [6]) NFR 5. FAIR principles (c.f. [20])		
b. Functional Requirements:		
Lists of requirements written as competency questions and sentences		
CQG1. from NLP algorithms	**CQG2.** from ISO 23386	**CQG3.** from LBD ontologies
CQ1.1. How can class labels be mapped to information elements? CQ1.2. How can numerical values be retrieved as boundary values for properties? CQ1.3. What provenance data can be extracted? CQ1.4. How are enumerations retrieved for properties? CQ1.5. Which physical characteristics and non-literal data is retrieved and modeled?	CQ2.1. What is a property? CQ2.2. What is a group of properties? CQ2.3. What is an interconnected data dictionary? CQ2.4. What is the provenance of an information element? CQ2.5 What are the relations between information elements?	CQ3.1. How can properties be assigned to building elements? CQ3.2. How can values be asserted for properties? CQ3.3. How can values of properties be changed over time? CQ3.4. How can measurements and units be represented? CQ3.5. How can properties be reused by other ontologies?

of defined elements (CQ1.4). The mapping of the enumeration values can be done via the ISO 23386 data model. More complex data types such as physical characteristics, non-literal data, and the retrieval and modeling of these are other important subjects that need to be addressed. Calculation formulas for complex physical values can be modeled within the standard as dynamic properties. Non-literal data types can be mapped directly within a property similar to enumeration types.

The prerequisites that are placed to the ontology by the NER can be ensured by the exact modeling of the attribute model, as can be seen from the previous considerations. Additionally, the selection of the label classes of the NER should be done with diligence and considering the listed prerequisites.

4.4.2. CQG2. Requirements from ISO 23386 As denoted in Section 2.2, the ISO 23886 specifies the attributes with which information elements can be defined from a building code and every attribute within the data model can be identified by a unique code. In addition, it maps processes that makes it possible to link different data catalogs into a network of the interconnected data dictionary. To map the data model denoted in ISO 23386 within the ontology, five different competence questions have to be addressed within the analysis of the functional requirements derived from the standard (c.f. Table 1, CQ2.1 to CQ2.5).

First, it must be classified what a property (CQ2.1) or property group (CQ2.2) is. The standard defines the property as the inherent or acquired characteristic of an object or item which can be assigned to several groups of properties [5]. A single property is described by up to forty different mandatory and optional attributes, which are also required to be represented within the ontology proposed in this paper. Groups of properties are defined as a collection enabling the properties to be prearranged or organized and can be classified into five different categories. Within the ontology to be developed, in addition to these stated category, up to 22 further attributes are to be provided for the description of a feature set [5].

The third competency question (CQ2.3) emphasizes determining the terminology of an interconnected data dictionary. An interconnected data dictionary refers to a set of complying data dictionaries that are linked by the respective attributes specified in ISO 23386. The required attributes are part of the attribute lists for information elements and are thus already covered by CQ2.1 and CQ2.2. Furthermore, the provenance (CQ2.4) of an information element needs to be captured within the ontology. The provenance is also taken into account within the attribution of information elements and is represented by several attributes. Finally, it must be clarified which relationships between information elements must be covered within the ontology (CQ2.5).

In conclusion, it can be stated that all requirements for the ontology, which arise from ISO 23386, have to be fulfilled through the exact mapping of the attribute tables listed in the standard.

4.4.3. CQG3. Requirements from existing ontologies Based on the review of existing ontologies describing properties in the field of LBD, five competency questions were defined to enable seamless reuse of properties according to this research in BIM workflows (c.f. Table 1, CQ3.1 to CQ3.5). The assignment of properties to building elements (CQ3.1) is an essential requirement to make this ontology usable. Therefore, the BOT and OPM were examined in detail providing different levels of property assignments to building elements or any FoI as a top-level approach to property management [8]. Considering the modeling of property attachment to FoI in other domains, for instance, in the BPO properties are attached to products using a `bpo:hasAttribute` property [17]. Thus, in this ontology concept, an object property needs to be provided having the property defined as the `rdfs:range` and allowing any FoI as `rdfs:domain`. The assertion of values to attached properties (CQ3.2) is demonstrated in OPM, DICO, and BPO. These ontologies reuse established patterns for the definitions of values using the `schema:value` object property[1] for literal values and the `schema:PropertyValue` type[2] for

[1] Schema.org value property: `https://schema.org/value`
[2] Schema.org PropertyValue class: `https://schema.org/PropertyValue`

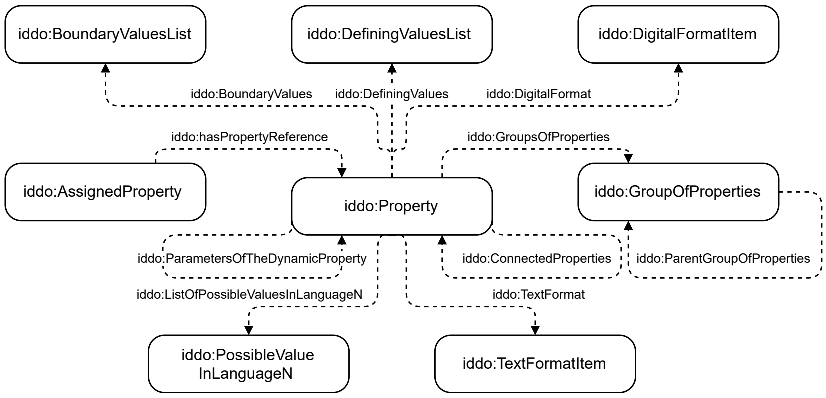

Figure 2: Ontology representation of the main classes including object properties

complex values with boundaries.

Addressing CQ3.3, the OPM can be utilized to model properties that change over time using the `opm:PropertyState` concept. However, ISO 23386 has defined a management process for the status of the property definition itself. The reviewed ontologies represent units according to CQ3.4 using the Quantities, Units, Dimensions, and Types (QUDT) unit catalog [23]. Therefore, all unit-related patterns in this ontology should match the QUDT unit definition. To make data catalogs according to this ontology reusable in BIM workflows (CQ3.5), these catalogs need to be published in accessible repositories and data catalog servers as shown in Figure 1. These repositories or servers themselves need to provide data catalogs according to fair principles and must fulfill the management rules defined by ISO 23386.

4.5. Non-Functional Requirements

Five Non-Functional Requirements (NFR) are identified during the requirements engineering (see Table 1). According to Costin and Eastman [6], concise terminology (NFR1) is required, which can be guaranteed during ontology development by taking the ISO 23386 as the reference data model. Furthermore, consistency (NFR2) must be adhered to in order to avoid duplicate information or overconstrained property assertions. The extendibility (NFR3) can be achieved by using standardized languages and existing concepts and patterns for the ontology development. Reliability (NFR4) is another important requirement for the long-term provision of the ontology and its metadata to enable others to work with it. Finally, the FAIR principles (NFR5) for scientific data management are an important requirement for ontologies to be findable, accessible, interoperable, and reusable. This sums up the NFRs and is especially important during ontology development, documentation, and publishing [20].

4.6. Requirements evaluation and validation

In general, all functional and non-functional requirements can be combined. Nevertheless, the strict implementation according to the ISO 23386 standardization and the flexible embedding in the generation process of data sets as well as the use in the BIM workflow employing LBD influence each other at certain points. Conformity to the data model from the standardization is determined by RDFS constraints during modeling and constantly checked afterwards with the help of SHACL so that the data records always comply with the standard.

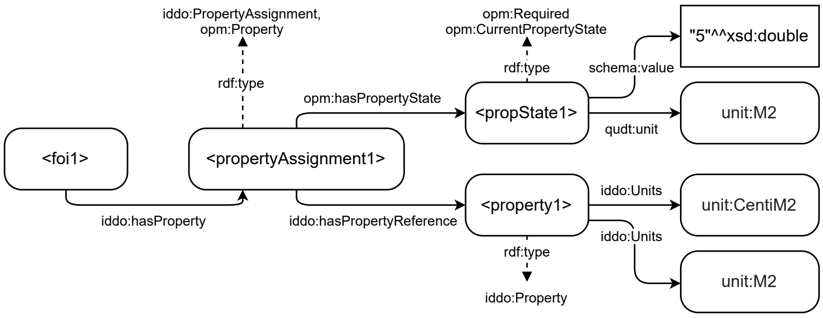

Figure 3: Assignment of a property to a FoI using a PropertyAssignment individual with property state and value, and the property reference with possible units

5. The Interconnected Data Dictionary Ontology (IDDO)

In the following, the Interconnected Data Dictionary Ontology (IDDO) is presented as developed based on the requirements analysis. The ontology maps the data model of the ISO 23386 and establishes additional classes for linking properties to an FoI. Figure 2 shows the three main ontology classes `iddo:Property`, `iddo:GroupOfProperties`, `iddo:AssignedPropety` and additionally a selection of descriptive classes and their respective object properties.

The allocation of properties to a property group is performed via the object property `iddo:GroupsOfProperty`. In addition, both information elements can refer to an individual of their class. This is essential for the class `iddo:GroupOfProperties` and is realized by the object property `iddo:ParentGroupOfProperties` to be able to create the mono-hierarchical classification and the inheritance structure between groups of properties as required by the standard. The two object properties `iddo:ParametersOfTheDynamicProperty` and `iddo:ConnectedProperty` provide two options to link properties together. The first one can be used to attach a set of properties to one dynamic property to enable the depiction of complex calculated parameters. The second one can be facilitated to relate properties that are related to each other. For the detailed description of all classes, objects, and data properties modeled in the IDDO, the ontology is fully documented[3] online.

During the development of the IDDO ontology, a particular focus was set on integration into the design workflow. Also, existing patterns, especially from the OPM ontology, were reused. Therefore, the assignment of an `iddo:Property` to a FoI is demonstrated in Figure 3. The FoI is attributed with an `iddo:hasProperty` object property to an individual of the two classes `iddo:PropertyAssignment` and `opm:Property`. This individual refers to a property state using the property state pattern of OPM ontology and to an `iddo:Property` using the `iddo:hasPropertyReference` object property. These paths are restricted so that a property assignment can only refer to exactly one property, but can be asserted to one or more property states according to OPM. The property state defines a value as well as a unit (if applicable). The `iddo:Property` property has been asserted by the required information, which is restricted as well using RDFS restrictions. In Figure 3, the complete extent of asserted properties has been left out for brevity. The figure shows that the property defines two possible units using the `iddo:Units` predicate and a QUDT unit individual. To define a value of a property as required, the property state of the property assignment can additionally be typed as `opm:Required`. Thus, the information can be established using IDDO properties according to the presented use cases 1 and 2.

In the next step, the assigned property can be utilized for rule checking and validation

[3] IDDO documentation: `https://w3id.org/iddo`

purposes as introduced in use case 4 (see Table 1). Therefore, in Listing 1, a property and a property assignment are represented in Turtle as referred to in Figure 3. SHACL shapes are defined as Turtle files using the SHACL vocabulary as shown in Listing 2.

The Turtle file contains a node shape that targets the class of assigned properties so that this shape will be applied on all instances of the `iddo:AssignedProperty` class. During validation, the examined instances are denoted as focus nodes. The shape is annotated with a message that is provided in case a focus node violates the shape constraints. To validate the assigned unit of a property value, this shape defines a SPARQL-based constraint. The constraint is written in SPARQL syntax and is applied to each focus node which is prebound to the variable `$this`.

Listing 1: data.ttl

```
dict:DIN277_NetFloorArea
  a iddo:Property ;
  iddo:GloballyUniqueIdentifier
    "0451dc04-6955-41ae-90d3-5ba6d090aa2c" ;
  iddo:NameInLanguage "Net floor area"@en ;
  [...]
  iddo:Units unit:CentiM2 ;
  iddo:Units unit:M2 ;
.

inst:NetFloorArea_scDw54vuxUuNnhJ6XHr
  a iddo:AssignedProperty ;
  opm:hasPropertyState
    [ a opm:CurrentPropertyState;
      qudt:unit unit:M3;
      schema:value "22.85"^^xsd:double ;
    ];
  iddo:hasPropertyReference dict:DIN277_NetFloorArea;
.
```

Listing 2: unit.shapes.ttl

```
[] a sh:NodeShape ;
  sh:targetClass iddo:AssignedProperty ;
  sh:message "The assigned unit of the property
    state is not allowed by the property." ;
  sh:sparql [
    a sh:SPARQLConstraint ;
    sh:select """
    PREFIX iddo: <http://w3id.org/iddo#>
    PREFIX qudt: <http://qudt.org/2.1/schema/qudt/>
    PREFIX opm:  <https://w3id.org/opm#>
    SELECT ?state ?assignedUnit ?property $this
    WHERE {
     $this opm:hasPropertyState ?state .
     ?state qudt:unit ?assignedUnit .
     $this iddo:hasPropertyReference ?property .
        FILTER NOT EXISTS { ?property iddo:Units
           ?assignedUnit}
    }""" ;
  ] .
```

The assigned unit of the property state is not allowed by the property.

Location: [Focus node] - [http://icdd.vm.rub.de/ontology/iddo/instances#NetFloorArea_scDw54vuxUuNnhJ6XHr]

Test: [Shape] - [http://icdd.vm.rub.de/ontology/iddo#AssignedPropertyUnitValidation] - [Value] - [http://icdd.vm.rub.de/ontology/iddo/instances#NetFloorArea_scDw54vuxUuNnhJ6XHr]

Figure 4: The failed property validation of `data.ttl` using the defined shape in `unit.shapes.ttl`

Within the SPARQL query, all focus nodes are validated whether their assigned unit is in the range of defined units of the property. The validation fails if no respective triple exists in the graph. For the given example data and shape file, the validation has been executed. The validation fails because `unit:M3` was assigned as a unit but the property only allows `unit:M2` and `unit:CentiM2` as possible units. This validation using IDDO properties shows the potential that validation brings for the usage in the building lifecycle. Further possible application cases are the validation of boundary values or possible enum values defined by properties.

6. Conclusions and further research

This paper provides a requirements analysis for the development of an ontology for representing and maintaining properties from an automated building code interpretation into interconnected data dictionaries. The scope, purpose, intended users, and use cases as well as functional and non-functional requirements were included and formalized. Non-functional requirements are considered from literature and the FAIR principles for data sharing. Functional requirements were systematically identified according to established methodologies and grouped into three groups of competency questions from (1) building code recognition, (2) standardization, and (3) existing ontologies to fulfill the multi-requirements analysis. The development process of the proposed ontology was thereby considerably improved. The result of the analysis and ontology development is the IDDO ontology. For clarification, the ontology documentation also includes the origin of all classes and properties from ISO 23386 with the corresponding codes.

The feasibility of the ontology concerning the requirements and use cases is given and is successfully demonstrated for the defined use cases 2, 3, and 4 (see Table 1). The property assignment is revised based on existing ontologies. To combine aspects from the OPM and the BPO ontologies, a proxy class `iddo:PropertyAssignment` was introduced. The demonstration shows how the IDDO can be used in the building lifecycle and how the quality assurance of properties can be conducted using SHACL shapes. Thus, the data structure shown can be used to create data catalogs based on building codes. An evaluation of the ontology as a data structure (use case 1) for automatic building code interpretation and as a semantic knowledge base is still pending and is the next step in the ongoing research.

References

[1] Pauwels P, Costin A and Rasmussen M H 2022 *Industry 4.0 for the Built Environment* Springer eBook Collection ed Bolpagni M, Gavina R and Ribeiro D (Cham: Springer International Publishing and Imprint Springer) pp 157–183

[2] Oraskari J, Beetz J and Senthilvel M 2021 *Proc. of the 38th CIB W78 Conf. on Information and Communication Technologies for AECO*

[3] Schönfelder P and König M 2021 *Proc. 38th International Symposium on Automation and Robotics in Construction (ISARC 2021)*

[4] Locatelli M, Seghezzi E, Pellegrini L, Tagliabue L C and Di Giuda G M 2021 *Buildings* **11** 583

[5] ISO 23386 2020-03 Building information modelling and other digital processes used in construction: Methodology to describe, author and maintain properties in interconnected data dictionaries

[6] Costin A and Eastman C 2017 *Proc. of the Joint Conf. on Computing in Construction (JC3)* 2017 Lean and Computing in Construction Congress (LC3) ed Bosché F, Brilakis I and Sacks R (Edinburgh, UK: Heriot-Watt University) pp 533–540

[7] Rasmussen M H, Pauwels P, Hviid C A and Karlshøj J 2017 *Proc. of the Joint Conf. on Computing in Construction (JC3)* 2017 Lean and Computing in Construction Congress (LC3) ed Bosché F, Brilakis I and Sacks R (Edinburgh, UK: Heriot-Watt University) pp 237–244

[8] Rasmussen M H, Lefrancois M, Bonduel M, Hviid C A and Karlshøj J 2018 *Proc. of the 6th Linked Data in Architecture and Construction Workshop (LDAC)* ed Poveda-Villalón M, Pauwels P and Roxin A

[9] Fuchs S and Amor R 2021 *Proc. of the 38th International Conf. of CIB W78*

[10] Moon S, Lee G, Chi S and Oh H 2021 *Journal of Construction Engineering and Management* **147**

[11] Alani Y, Dawood N, Rodriguez S and Dawood H 2020 *Proc. 37th CIB W78 Conf. 2020* CIB W78 Proc. ed Toledo Santos E and Scheer S pp 141–155

[12] buildingSMART 2021 bSDD buildingSMART Data Dictionary (bSDD) URL https://github.com/buildingSMART/bSDD

[13] Beetz J, van Leeuwen J and de Vries B 2009 *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **23** 89–101

[14] Schneider G F 2017 *Proc. of the 5th Linked Data in Architecture and Construction Workshop (LDAC) 2017* ed Roxin A

[15] Bonduel M, Oraskari J, Pauwels P, Vergauwen M and Klein R 2018 *Proc. of the 6th Linked Data in Architecture and Construction Workshop (LDAC)* vol 2159 ed Poveda-Villalón M, Pauwels P and Roxin A pp 34–43

[16] Zheng Y, Törmä S and Seppänen O 2021 *Automation in Construction* **132** 103930

[17] Wagner A, Sprenger W, Maurer C, Kuhn T E and Rüppel U 2022 *Automation in Construction* **133** 103927

[18] Suárez-Figueroa M C and Gómez-Pérez A 2012 Ontology Requirements Specification *Ontology Engineering in a Networked World* (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 93–106

[19] Poveda-Villalón M, Fernández-Izquierdo A, Fernández-López M and García Castro R 2022 *Engineering Applications of Artificial Intelligence* **111** 104755

[20] Poveda-Villalón M, Espinoza-Arias P, Garijo D and Corcho O 2020 *Knowledge Engineering and Knowledge Management* (*Lecture Notes in Computer Science* vol 12387) ed Keet C M and Dumontier M (Cham: Springer International Publishing) pp 255–270

[21] Bezerra C, Freitas F and Santana F 2013 *2013 IEEE/WIC/ACM International Joint Conf. on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (IEEE) pp 284–285

[22] Motik B, Patel-Schneider P and Parsia B 2012 OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) URL https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/

[23] QUDT 2015 Quantities, Units, Dimensions and Types URL http://qudt.org/2.1/schema/qudt