Integration of BIM-related bridge information in an ontological knowledgebase

Al-Hakam Hamdan and Raimar J. Scherer

Institute of Construction Informatics,
Technische Universität Dresden, Dresden, Germany
Al-Hakam.Hamdan@tu-dresden.de

Abstract. Currently, utilizing digital representations of bridge constructions is still limited to geometry-based models with none or only little semantic data. Consequently, assessing these models requires the interpretation of external sources, e.g. relational databases or reports. Despite new approaches in the Building Information Modeling (BIM) domain such as the IFC-Bridge extension for the Industry Foundation Classes (IFC) try to provide models where geometric and semantic data are combined, a great proportion of information from current and future domains that are relevant for the bridge industry are not covered. For this reason, an extensible web ontology inspired from the Building Topology Ontology (BOT) for buildings has been developed, which functions as a core ontology for bridge representations and therefore covers all necessary general information used in this domain. The developed core ontology is interlinked with multiple domain specific bridge extensions in a bridge ontology framework that is applied on a test scenario. In this paper, the components of the bridge ontology framework are explained as well as the application on the test case. In addition, ontology alignments for BOT and ifcOWL are proposed as well as shapes for ontology validation. Furthermore, the functionality of a developed software prototype is described that generates the bridge ontology from a given IFC model.

Keywords: Bridge Model, Building Information Modeling, Industry Foundation Classes, Ontologies, Linked Data

1 Introduction

Semantic Web technologies are increasingly used for representing buildings digitally in architecture, engineering and construction (AEC) and complement existing model approaches and applications in Building Information Modelling (BIM). However, most of the current approaches and researches focus solely on the building sector, which makes it difficult to apply them to other construction types. In this regard, one domain that could particularly benefit from utilization of semantic representations is the bridge sector. This is especially the case because the maintenance of existing bridges often only involves the acquisition of semantic data, whereby geometry-based models are rarely used. Utilizing ontologies from the building domain for semantic bridge representations is technically possible, however it would still promote misunderstandings due to a building-focused terminology. Furthermore, certain terms that are common in the bridge sector are not supported by the building ontologies. For this reason, a modular ontological framework has been developed in the research, presented in this paper that can be used for representing bridge information. The framework consists of a core ontology named Bridge Topology Ontology (BROT) that has been prototypically developed in a previous research [1] and functions as the bridge counterpart of the Building Topology Ontology (BOT) [2]. In this regard, several concepts from BOT are reused in BROT. Although, it would be possible to just extend BOT with additional terminological components for bridge representation, the current class and property definitions of BOT are targeted solely on buildings. Thus, most bridge-related additions to BOT could not relate to the existing BOT components, e.g. through class or property inheritance, without contradicting the provided definitions. An alternative solution would be the development of a generic construction ontology, which merges the concepts of BOT together with ontology extensions that can be used for other construction types. Nonetheless, also in this solution the usage of a bridge specific ontology (as extension) would be necessary.

The proposed framework consists of multiple ontology extensions to allow the definition of more specific information, e.g. classifying specific bridge components or material parameters.

The web ontologies developed in this research utilize Linked Data and therefore web standards such as the Resource Description Framework (RDF) and Web Ontology Language (OWL). In this paper, the terminological structure of these models is described in detail. Furthermore, the developed web ontologies are applied on a test case, where a semantic representation of an existing bridge is modeled. Thereby, a tool is utilized, which has also been developed in this research and allows for the transformation of a BIM model serialized using Industry Foundation Classes (IFC) into a web ontology. The ontologies and test scenario are stored on Github[1]. The bridge ontologies are also accessible through the website of the research project wiSIB[2].

Table 1 contains an overview of the used prefixes throughout this research paper.

Prefix	Namespace
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
sh	http://www.w3.org/ns/shacl#
bot	https://w3id.org/bot#
brot	https://w3id.org/brot#
brcomp	https://w3id.org/brcomp#
bridge	https://w3id.org/bridge#
brstr	https://w3id.org/brstr#
bmat	https://w3id.org/bmat#
ifc	http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD1#

Table 1. Namespaces and prefixes used in this paper

[1] https://github.com/Alhakam/bridgeOntology

[2] https://www.wisib.de/

2 Related Work

A great proportion of current research focuses on conceptualizing appropriate data structures for bridge representations. Several drafts for an IFC extension that covers the bridge domain were proposed, the most prominent ones being the French-Japanese collaboration project IFC-Bridge [3] and BrIM [4]. Furthermore, an IFC 4.2 draft where an IFC-Bridge extension is proposed has been published as result of a standardization project, supported by buildingSmart and several international ministries of transport [5]. At the publication of this paper the draft has been publicly available for review and comment. The classes and attributes defined in these extensions mainly cover only information used in the design phase and other lifecycle phases are not or only rudimentarily supported. This is especially a problem in the bridge domain, where the maintenance phase is of greater significance than in the building domain and periodic inspections are mandatory, due to the atmospheric conditions and heavy traffic to which the structures are exposed. Furthermore, contrary to more modular data structures, e.g. Linked Data models, the monolithic structure of IFC leads to a slower approval process of new extensions in the main schema and consequently due to its inflexibility the solely use of IFC models for bridge representations will remain impractical when using unsupported domains.

Currently, bridge management systems are used frequently in practice for managing information recorded during the bridge lifecycle. In general, they are built around one or multiple relational databases that store various bridge information as well as the related metadata. Often these databases are developed on the foundation of national or international standards, such as SIB-Bauwerke [9], which is based on the German standard DIN 1076 [6]. However, the data in these databases are often not linked with elements in other relevant models, e.g. the IFC model or refurbishment plans. Consequently, the information from various models and documents and its context must be interpreted manually by human experts, which is usually an error prone task. Furthermore, the databases lack the flexibility of other data models, e.g. an OWL ontology, since the inserted data must follow a specific non-standardized scheme, which is often only supported by one proprietary software and difficult to extend.

Several approaches have been developed that utilize the benefits of Linked Data for bridge information management. Helmerich developed an ontological systems called BRONTEX [7] for managing knowledge about existing bridges. The system allows for defining material and structural parameters and interlinking them to a bridge test case as well as the assignment of defects and methods for detecting these defects. In this regard, the ontology functions as a knowledge base for storing expert knowledge as well as related case studies, which can be queried via SPARQL. However, the ontology is not suited towards representing a bridge construction due to the lack of construction specific information, e.g. topology or the representation of bridge components. An ontology for representing existing bridges entitled as “Bridge Maintenance Ontology” has been developed by Ren et al. [8]. Thereby, various bridge components can be defined and assigned to a bridge structure, e.g. the bridge deck or superstructure. Furthermore, maintenance relevant information such as detected damages, events or processed refurbishment solutions can be added. However, all terminological components (TBox) that define domain specific information are not structured in a modular ontology, which complicates the addition of other ontologies and the extension process itself. Object properties for defining topological relations, such as the adjacency or aggregation of bridge components are missing. Furthermore, bridge components are not structured in hierarchical superclasses and therefore more difficult to query (e.g. querying all building components of the bridge deck). Besides the aforementioned ontological approaches that are independent from any standards, an ontology for bridge representation based on the SIB-Bauwerke database is under development. Similar to the ontology by Ren et al. the TBox is structured monolithically. Furthermore, the ontology components were modeled accurately according to the elements from the relational database from SIB-Bauwerke [9].

3 Bridge Ontology Framework

The developed bridge ontology is structured modularly, consisting of the core ontology BROT that covers general essential classes and properties for representing a bridge as well as multiple extensions, which provide domain specific information and contain data that are often not relevant for every stakeholder (see **Fig. 1**). In this regard, the ontology framework has been developed according to the methodical principle of ontology modularization, whereby a semantics-driven structure is utilized [10].

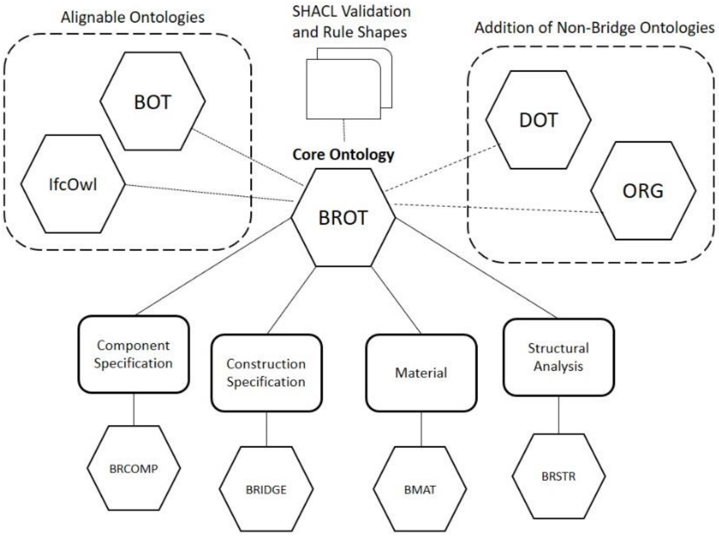


Fig. 1. Ontological Components of the Bridge Ontology Framework

The web ontology and all compatible extensions are developed in OWL. In the current prototypical status of this ontology framework the extensions consist of various web ontologies for further classification and characterization of bridge structures and their components, an ontology for material specification and an ontology for structural classification and properties. Additionally, web ontologies that are used outside the bridge domain like the Damage Topology Ontology (DOT) [11] or the Organization Ontology (ORG) [12] can be integrated in the ontology framework. In this regard, DOT is compatible with the core ontology BROT, i.e. several object properties already relate to BROT instances. Furthermore, ontology alignments have been developed for existing AEC ontologies, namely BOT [2] and IfcOwl [13]. Thus, existing web ontologies that are compatible with these two ontologies, can also be reused for BROT.

3.1 Structure of the Core Ontology BROT

The Bridge Ontology Topology (BROT) uses several existing concepts that are already used in BOT and therefore the web ontology can be viewed as a bridge counterpart of BOT (see **Fig. 2**). Like in BOT, a distinction between zones and contained elements is made. In this regard, zones are classified as brot:SpatialZone, while elements that are contained in a specific zone use brot:Component. The topology of the bridge structure can be defined via object properties.

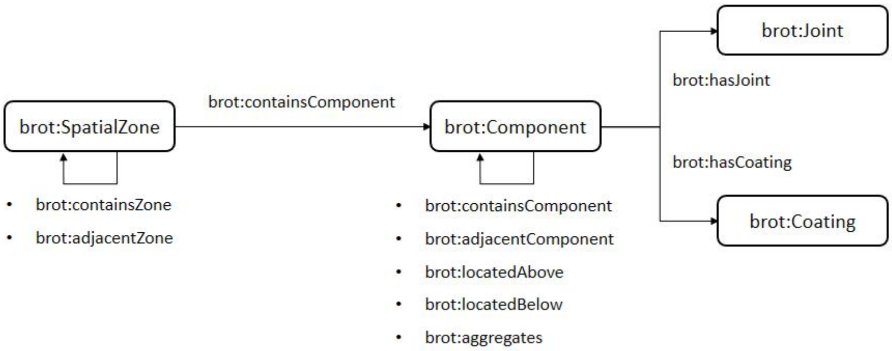


Fig. 2. General classes and object properties of BROT

In this regard, the containment of one brot:SpatialZone instance nested in another is defined through the transitive property brot:containsZone. Adjacencies between zones can be described via the symmetric property brot:adjacentZone. Similar principles are applied to instances of brot:Component, whereby the transitive property brot:aggregates defines the relation between a component and its aggregated subcomponent and the symmetric property brot:adjacentComponent the adjacency between two components. If an instance of brot:SpatialZone contains a brot:Component object, this relation can be defined via brot:containsComponent. In addition, by applying a reasoning engine to the ontology, it can be derived through a property chain axiom that instances of brot:Component that are aggregated in another brot:Component instance are also contained in its brot:SpatialZone instance via brot:containsZone. The same applies to instances of brot:Component that are contained in a brot:SpatialZone instance via brot:containsComponent, which is contained in another brot:SpatialZone instance through brot:containsZone. Consequently the brot:Component instance would be reasoned as member of both instances of brot:SpatialZone. Due to the more standardized structure of bridges compared to buildings, it is also feasible to describe the vertical topology of components. Thereby, the property brot:locatedAbove describes the relationship between two instances of brot:Component where the subject is located above the object in relation to the building ground. The inverse property therefore is brot:locatedBelow. Additionally, elements that supplement the components and thus cannot be modelled without a related component are defined through separate classes. Particularly, this is true for joints and coatings that belong to a component. Therefore, the classes brot:Joint and brot:Coating have been implemented and can be linked via the object properties brot:hasJoint and brot:hasCoating.

Besides these generic classes to model a semantic bridge representation, more specific classes are provided in the BROT terminology that are based on common and standardized terms. Thus, the class brot:SpatialZone possesses numerous subclasses that should be used in a specific pattern for modeling an appropriate bridge representation. Thereby, the recommended ontological bridge model should contain at least one instance of brot:Site, which represents the site or area where the construction is located. One or more instances of brot:Bridge should be assigned to the brot:Site instance. In this regard, it is not unusual to split bridge lanes that count as separate bridges from a structural point of view into multiple instances of brot:Bridge. Furthermore, each instance of brot:Bridge should consist of exactly one instance of brot:SubStructure and brot:SuperStructure. These two classes are based upon the commonly known terms of substructure and superstructure that group components in the bridge structure. Since all the aforementioned classes are subclasses of brot:SpatialZone, the relations are defined via brot:containsZone. Furthermore brot:SubStructure and brot:SuperStructure should be related to each other via brot:adjacentZone. The described pattern can be validated by using a SHACL shape.

3.2 BROT Extensions

The modular structure of the proposed bridge ontology aims for utilizing BROT as core ontology, which covers only information that are applicable to any type of bridge. Therefore, four additional ontologies have been developed as compatible extensions for BROT to cover more specific information. It should be noted that these extensions are not final and have been mainly designed in the scope of the project wiSiB to cover the given bridge test subject.

The Bridge Components Ontology (BRCOMP) provides classes and data properties for specifying bridge components. Thereby, BRCOMP adds four additional subclasses of brot:Component that are based on common engineering terms and characterize specific component groups which are often subject of queries. In this regard, brcomp:SubstructureComponent and brcomp:SuperstructureComponent refer to subclasses of brot:Component of which instances are always member of either brot:SubStructure or brot:SuperStructure. The class brcomp:FinishingComponent is used for representing components that serve no structural purpose, while brcomp:ReinforcingComponent refers to all instances that represent reinforcement components, such as reinforcing bars or tendons. For these classes, additional subclasses like brcomp:Abutment or brcomp:Pier are defined to provide a taxonomy for classifying bridge components. Furthermore, data properties for component characterization are available that are based on attributes used in the German DIN 1076 [6], e.g. brcomp:concreteCover or brcomp:installationYear. Similar to BRCOMP, the Bridge Classification Ontology (BRIDGE) provides subclasses and properties for a more detailed characterization of the entire bridge representation that is instantiated as brot:Bridge. For instance, classes such as bridge:SteelBridge or bridge:PrestressedConcreteBridge classify the bridge representation according to its material. Also the data properties support a great proportion of attributes that are used in inspection reports according to the German DIN 1076 [6], such as maximum construction height (bridge:maxConstructionHeight) or the construction year (bridge:constructionYear). For material specification the Building Material Ontology (BMAT) is used that can also be applied for other construction types, such as buildings as it has previously been done for natural stone facades in [14]. For characterizing instances based on their function in a structural analysis model, the Bridge Structure Ontology (BRSTR) has been developed. It provides a class for defining load bearing components (brstr:LoadBearingComponent) as subclass of brot:Component as well as a class for defining bridge spans (brstr:Span) which is a subclass of brot:SpatialZone. Furthermore, transitive object properties for load transfer are provided. Thereby, in the brstr:loads relation the subject transmits loads to the object. The inverse property therefore is brstr:isLoadedBy. A subproperty of brstr:loads is brstr:restsOn which refers to subjects that not only transmits loads to its object but also has direct physical contact with it. The inverse property therefore is brstr:bears. Besides the load transmission properties, several data properties are also available that describe structural parameters such as the degree of freedoms or support height.

3.3 Alignment between BROT and existing web ontologies

Since BROT is based on classes and properties already used in BOT, the usage of an alignment enhances the exchange between ontologies that use these terminologies. Furthermore, existing ontologies that are already used in conjunction with BOT are compatible with BROT, when applying the alignment presented in **Table 2**.

Subject	**Predicate**	**Object**
bot:Element	owl:equivalentClass	brot:Component
bot:Zone	owl:equivalentClass	brot:SpatialZone
bot:containsZone	owl:equivalentProperty	brot:containsZone
bot:adjacentZone	owl:equivalentProperty	brot:containsZone
bot:containsElement	owl:equivalentProperty	brot:containsElement
bot:hasSubElement	owl:equivalentProperty	brot:aggregates

Table 2. Proposed Alignment of BOT and BROT

Additionally, an alignment between BROT and IfcOWL is proposed in **Table 3**. In this regard, the IFC conversion tool described in chapter 4.1 is also based on this alignment. There, several classes that are defined exclusively for buildings are reinterpreted, e.g.

ifc:IfcBuilding is equivalent to brot:Bridge. Consequently, the proposed alignment should only be used in scenarios where the user is sure, that the ifcOWL ontology represents a bridge structure.

Subject	Predicate	Object
ifc:IfcSite	owl:equivalentClass	brot:Site
ifc:IfcBuilding	owl:equivalentClass	brot:Bridge
ifc:IfcBuildingElement	owl:equivalentClass	brot:Component
ifc:IfcSpatialStructureElement	owl:equivalentClass	brot:SpatialZone

Table 3. Proposed Alignment of BROT and IfcOwl

Relationships that describe aggregations, such as ifc:IfcRelAggregates or ifc:IfcRelDecomposes could not be defined as equivalent to brot:aggregates in OWL, since they are classes and not object properties. In OWL, object properties can only be defined as equivalent to other object properties.

4 Application of the Ontology Framework

The proposed ontology framework has been tested on an existing bridge construction. Beforehand, an IFC model of the bridge has been created by using a CAD tool. In a first step, information from the IFC model should be used for generating an ontological model using the BROT terminology. Since the IFC model exported from the CAD tool mainly contains geometrical data, additional semantic data are added manually using an ontology editor. The resulting ontological bridge representation should then be validated against SHACL shapes to ensure a correct topological structure.

4.1 Ontology Generation from IFC

In the scope of this research, a tool has been developed that allows for the generation of a bridge ontology based on the data from an IFC model. Besides the benefit of the automatized creation of a first core topology that can be semantically enriched in subsequent steps when using such a tool, it also allows for an immediate linking of IFC and OWL data according to ISO 21597 [15]. Thus, the generated ontology is embedded in an ICDD Multimodel [16] together with the IFC model and respective linksets. A comparable approach of dual IFC and OWL building models has been developed by Baumgärtel & Scherer [17] for establishing an intelligent virtual lab for the purpose of energy-efficient building design.

The IFC to BROT conversion is based on the alignment proposed in **Table 3**. Thereby, the tool analyses all existing data of a given IFC model and filters for entities of IfcBuildingElement for creating OWL instances of brot:Component. The same is done for entities of IfcRelationship to generate the related object properties. Furthermore, the recommended zone topology according to chapter 3.1 is generated and fitting instances are assigned to the new instances of brot:SuperStructure and brot:SubStructure via brot:containsElement. If the IFC model contains either a single entity of IfcSite or if the IfcBuildingElement entities are related to the IfcSite entity (e.g. via IfcRelDecomposes) a corresponding brot:Site instance is created. The same principle is applied for IfcBuilding to generate a brot:Bridge instance. Regarding the generation of object properties related to IfcRelationship, only aggregation properties (brot:containsZone and brot:aggregates) based on IfcRelDecomposes or its subclass IfcRelAggregates are supported by the tool.

Since standards for bridge modeling in IFC such as IFC-Bridge are still in a draft state and therefore currently not supported by CAD tools, a generation of instances with more specific class definitions from BRCOMP or BRIDGE is not possible. However, manual annotations to IfcBuildingElement entities have been created using the BIM-Annotator by [18], so that it would be possible for the tool to filter these entities and assign BRCOMP classes to each generated instance based on the corresponding annotation. In this regard, the annotations are not assigned directly to the entities in the IFC model, e.g. by using the attribute ObjectType or Tag. Instead a separate annotation file is created that contains all relevant annotation properties and is linked via a Linkmodel with the IFC model based on the Multimodel approach by [19]. If the annotation Multimodel is present, the developed BROT generation tool can use it for generating more specific BRCOMP instances, e.g. brcomp:Abutment or brcomp:Pier. An appropriate assignment of the new instances to either brot:SubStructure or brot:SuperStructure can also be optionally processed. (Fig. 3) presents a section of the edited bridge ontology. Thereby, only a small proportion of the created classes and properties is presented, to show the principle in an understandable way.

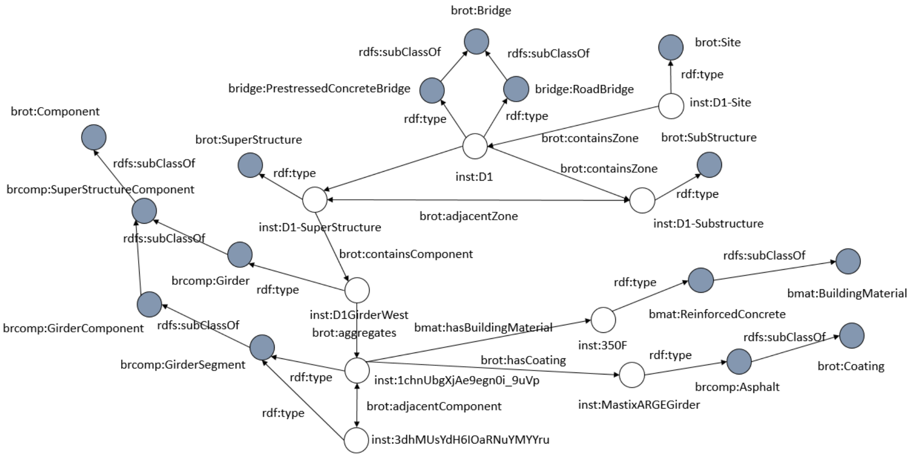
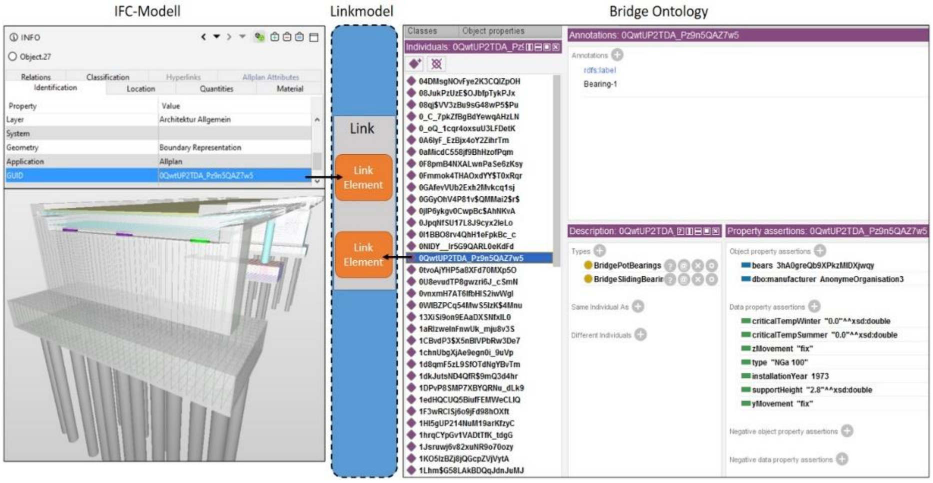


Fig. 3. Extract of the assertional components of the bridge ontology test case (simplified)

When generating each BROT instance, an icdd:LinkElement instance is also generated in parallel that links the BROT instance with the corresponding IFC entity. The resulting Linkset is then stored together with the IFC model and generated ontology in an ICDD[15].



Property	Value
Layer	Architektur Allgemein
System	
Geometry	Boundary Representation
Application	Allplan
GUID	0QwtUP2TDA_Pz9n5QAZ7w5

Individuals: 0QwtUP2TDA_Pz

04DMsgNOvFye2K3CQIZpOH
08JukPzUzE$OJbfpTykPJx
08qj$VV3zBu9sG48wP5$Pu
0_C_7pkZfBgBdYewqAHzLN
0_oQ_1cqr4oxsuU3LFDetK
0A6lyF_EzBjx4oY2ZihrTm
0aMicdC558jf9BhHzofPqm
0F8pmB4NXALwnPaSe6zKsy
0Fmmok4THAOxdYY$T0xRqr
0GAfevVUb2Exh2Mvkcq1sj
0GGyOhV4P81v$QMMai2$r$
0jIP6ykgv0CwpBc$AhNKvA
0JpqNfSU17L8J9cyx2IeLo
0l1BBO8rv4QhH1eFpkBc_c
0NIDY__lr5G9QARL0eKdFd
0QwtUP2TDA_Pz9n5QAZ7w5
0tvoAjYHP5a8XFd70MXp5O
0U8evudTP8gwzri6J_cSmN
0vnxmH7AT6IfbHIS2iwWgl
0WIBZPCq54MwS5IzK$4Mnu
13XiSi9on9EAaDXSNfxIL0
1aRIzweInFnwUk_mju8v3S
1CBvdP3$X5nBIVPbRw3De7
1chnUbgXjAe9egn0i_9uVp
1d8qmF5zL9SfOTdNgYBvTm
1dkJutsND4QfR$9mQ3d4hr
1DPvP8SMP7XBYQRNu_dLk9
1edHQCUQ5BiufFEMWeCLIQ
1F3wRCISj6o9jFd98hOXft
1HI5gUP214NuM19arKfzyC
1hrqCYpGv1VADtTfK_tdgG
1Jsruwj6v82xuNR9o70ozy
1KO5IzBZj8jQGcpZVjVytA
1Lhm$G58LAkBDQqJdnJuMJ

Annotations: 0QwtUP2TDA_Pz9n5QAZ7w5

rdfs:label
Bearing-1

Description: 0QwtUP2TDA

BridgePotBearings
BridgeSlidingBearir

Property assertions: 0QwtUP2TDA_Pz9n5QAZ7w5

bears 3hA0greQb9XPkzMIDXjwqy
dbo:manufacturer AnonymeOrganisation3

criticalTempWinter "0.0"^^xsd:double
criticalTempSummer "0.0"^^xsd:double
zMovement "fix"
type "NGa 100"
installationYear 1973
supportHeight "2.8"^^xsd:double
yMovement "fix"

Fig. 4. IFC Model linked with the generated bridge ontology

The current prototype of the IFC-To-BROT transformation tool only generates an ontology, which contains topological information as well as some class assignments. A great amount of relevant information, e.g. from inspection reports are not implemented. Therefore, it is recommended to add further semantic data to the resulting ontological model, utilizing more specific terminologies from compatible extensions in a subsequent step. Thus, in the test case, used in this research, the generated ontology has been semantically enriched using an OWL editor tool.

4.2 SHACL Validation for BROT

When using the ontology framework and especially the core ontology BROT for modeling a bridge representation, some classes and properties should be modeled in defined structures that generally reflect the hierarchical topology of every bridge. For instance, it should not be allowed to model the triple "brot:Bridge brot:containsZone brot:Site" as a bridge construction never contains a construction site. However, to prevent such triple constellations in the TBox definition, concepts must be added that limit the overall modeling capability of the ontology, such as removing the generic object property brot:containsZone and instead implement more specific properties, e.g. brot:containsBridge (whereby the subject must be brot:Site and object brot:Bridge).

Through using validation constraints, that are defined outside of the ontology, this problem can be partially solved, by recommending the optional use of these constraints alongside the ontology. Thereby, the Shapes Constraint Language (SHACL), which is a W3C standard has been used in this research in order to define validation shapes in a separate file. In the following section, several SHACL shapes for validating a well formed BROT ontology structure are presented. For testing the SHACL shapes, the online tool SHACL Playground[3] has been used.

[3] https://shacl.org/playground/

Shape 1: According to the recommended zone structure discussed in chapter 3.1 each instance of brot:Site should contain at least one instance of brot:Bridge. Therefore, a node shape (see **Listing *1***) refers to the class brot:Site and has also a property shape assigned where the mentioned constrained is defined. In this regard the required predicate (brot:containsZone) and object (brot:Bridge) are defined as well as the cardinality (sh:minCount 1). Furthermore, the node kind has been defined as IRI, since brot:containsZone is an object property and thus literals are not allowed.

```
brot:Site
        a rdfs:Class , sh:NodeShape ;
        sh:property [
                rdf:type sh:PropertyShape ;
                sh:path brot:containsZone ;
                sh:class brot:Bridge ;
                sh:minCount 1 ;
                sh:nodeKind sh:IRI ;
        ] ;
.
```

Listing 1. Validation Shape for checking if brot:Site contains at least 1 brot:Bridge

Shape 2: Each instance of brot:Bridge should contain exactly one instance of brot:SubStructure and one instance of brot:SuperStructure. Furthermore, no instance of brot:Site should be assigned to brot:Bridge. The required constraints are shown in the node shape presented in **Listing *2***.

```
brot:Bridge
        a rdfs:Class , sh:NodeShape ;
        sh:property [
                rdf:type sh:PropertyShape ;
                sh:path brot:containsZone ;
                sh:class brot:SubStructure ;
                sh:minCount 1 ;
                sh:maxCount 1 ;
                sh:nodeKind sh:IRI ;
        ] ;
        sh:property [
                rdf:type sh:PropertyShape ;
                sh:path brot:containsZone ;
                sh:class brot:SuperStructure ;
                sh:minCount 1 ;
                sh:maxCount 1 ;
                sh:nodeKind sh:IRI ;
        ] ;
        sh:not [
                sh:property [
                rdf:type sh:PropertyShape ;
                sh:path brot:containsZone ;
```

```
                               sh:class brot:Bridge ;
                               sh:minCount 1 ;
                               ]
                     ]
           .
```

Listing 2. Validation Shape that defines assignable brot:SpatialZone instances for brot:Bridge

Shape 3: An instance of brot:SuperStructure must be connected to exactly one instance of brot:SubStructure via the object property brot:adjacentZone. Furthermore, no instances of the classes brot:Site, brot:Bridge or brot:SubStructure should be contained in it via the property brot:containsZone. A similar shape for brot:SubStructure has been defined, whereby brot:SuperStructure and brot:SubStructure definitions have been exchanged. **Listing *3*** shows the node shape for brot:SuperStructure.

```
        brot:SuperStructure
                a rdfs:Class , sh:NodeShape ;
                sh:property [
                        rdf:type sh:PropertyShape ;
                        sh:path brot:adjacentZone ;
                        sh:class brot:SubStructure ;
                        sh:minCount 1 ;
                        sh:maxCount 1 ;
                        sh:nodeKind sh:IRI ;
                ]
                sh:not [
                        sh:property [
                        rdf:type sh:PropertyShape ;
                        sh:path brot:containsZone ;
                        sh:or (
                                [sh:class brot:Site ;]
                                [sh:class brot:Bridge ;]
                                [sh:class brot:SubStructure ;]
                        ) ;
                        sh:minCount 1 ;
                        ]
                ] ;
        .
```

Listing 3. Validation Shape for brot:SuperStructure

In this regard, additional shapes for the BROT extensions have been developed. For instance, constraints that refer to BRCOMP instances have been created, whereby instances of brcomp:SubStructureComponent should not be contained in instances of brot:SuperStructure via the property brot:containsComponent.

Furthermore, besides validating existing ontological models, SHACL shapes can also be used for defining inference rules. Therefore, it would be possible to add missing information in the ontology, e.g. assigning an instance of brcomp:SubStructureComponent to the only existing brot:SubStructure of a brot:Bridge instance, even if the required property brot:containsComponent is not defined. However, this requires a previous validation of the ontology, to ensure that not multiple instances of brot:SubStructure are contained in a brot:Bridge instance.

5 Conclusions and Future Work

In this research, an ontology framework for semantic bridge representation is proposed. It consists of a core ontology named BROT, which provides a topological terminology similar to BOT for buildings[2]. By using additional web ontologies as extensions, more specific information such as taxonomies for bridges and their components or material parameters can be implemented in the bridge representation. Therefore, four extensions have been developed and applied on a test case, that provide classes and properties for characterizing general properties of the bridge itself (BRIDGE) and its components (BRCOMP) as well as for defining material information (BMAT) and information related to structural analysis (BRSTR). The ontology is also compatible with other web ontologies such as DOT for adding damage representations. Furthermore, alignments for BOT and IfcOWL have been proposed. In addition, a tool for generating a BROT ontology from an existing IFC model has been developed and applied on a test case. Thereby, the tool utilizes annotations in the IFC model to generate more specific classified instances. Furthermore, the resulting ontology is linked with the IFC model in an ICDD. It is still subject of future research, of how the developed ontology framework and tools can be used to solve specific problems that currently occur when working with models that contain no semantic data or interlinked information. Especially in the domain of existing bridges, the proposed ontologies can be of further use, where multiple information sets are generated during inspections and need to be linked with an appropriate semantic representation, especially since often no geometrical BIM model are available. Furthermore, the addition of a general construction ontology can enhance the compatibility between BROT and BOT and perhaps can support other construction domains as well, thus additional construction specific ontologies do not necessarily need to be developed in order to create semantic representations.

Acknowledgements

This research work was enabled by the support of the Federal Ministry of Education and Research of Germany through the funding of the projects wiSiB (project number 01|S16031C) and cyberBridge (project number 01|S16009B).

References

[1] T. Kozak and A.-H. Hamdan, "An ontological model for bridge representation" (Original in German: “Ein ontologisches Modell zur Repräsentation von Brücken,”) *Forum Bauinformatik 2019*, pp. 1–8, 2019.

[2] M. H. Rasmussen, P. Pauwels, C. A. Hviid, and J. Karlshøj, “Proposing a Central AEC Ontology That Allows for Domain Specific Extensions,” *Lean*

Comput. Constr. Congr. - Vol. 1 Proc. Jt. Conf. Comput. Constr., no. July, pp. 237–244, 2017.

[3] N. Yabuki and Z. Li, “International Collaboration for Developing the Bridge Product Model ‘IFC-Bridge,’” 2006.

[4] T. Chipman, “Bridge Information Model Standardization VOLUME III: COMPONENT MODELING,” *BIM-Infra*, vol. III, no. April, 2016.

[5] BuildingSMART, “Version 4.2 Draft - IFC Bridge proposed extension.” 2019.

[6] DIN Deutsches Institut für Normung e.V., "Building inspection according to DIN 1076", “Bauwerksprüfung nach DIN 1076,” 1999.

[7] R. Helmerich, “Knowledge representation system about existing bridges,” in *Bridge Maintenance, Safety, Management, Resilience and Sustainability*, Biondini and Frangopol, Eds. Taylor & Francis Group, London, 2012

[8] G. Ren, R. Ding, and H. Li, “Building an ontological knowledgebase for bridge maintenance,” *Adv. Eng. Softw.*, vol. 130, no. July 2018, pp. 24–40, 2019.

[9] “ASB-ING Condition,” 2017. [Online]. Available: https://roadotl.eu/static/eurotl-ontologies/testontologies/Germany/asb-ing-condition_doc/index-en.html. [Accessed: 16-Mar-2020].

[10] C. Parent and S. Spaccapietra, “An Overview of Modularity,” in *Modular Ontologies*, H. Stuckenschmidt, C. Parent, and S. Spaccapietra, Eds. Springer, 2009, pp. 5–23.

[11] A. Hamdan, M. Bonduel, and R. J. Scherer, “An ontological model for the representation of damage to constructions,” *7th Linked Data Archit. Constr. Work.*, 2019.

[12] D. (Epimorphics L. . Reynolds, “The Organization Ontology,” 2014.

[13] P. Pauwels and W. Terkaj, “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology,” *Autom. Constr.*, vol. 63, pp. 100–133, Mar. 2016.

[14] M. K. Seeaed and A. Hamdan, “BIMification of stone walls for maintenance management by utilizing Linked Data,” *31st Forum Bauinformatik*, no. November, 2019.

[15] ISO 21597-1: Information container for linked document delivery - Exchange specification - Part 1: Container,” 2020.

[16] R. J. Scherer and P. Katranuschkov, “Context capturing of Multi Information Resources for the Data Exchange in Collaborative Project Environments,” in *Proceedings of the European Conference on Computing in Construction (EC3 2019)*, 2019.

[17] K. Baumgärtel and R. J. Scherer, “Automatic ontology-based green building design parameter variation and evaluation in thermal energy building performance analyses,” *eWork Ebus. Archit. Eng. Constr. - Proc. 11th Eur. Conf. Prod. Process Model. ECPPM 2016*, pp. 667–672, 2016.

[18] A. Ismail, Y. Srewil, R. Scherer, and T. Mansperger, “Semantic Enrichment and Multimodel Data Exchange Approach for CFD Analysis of Bridges,” *EG-ICE Work.*, 2016.

[19] S. Fuchs and R. J. Scherer, “Multimodels — Instant nD-modeling using original data,” *Autom. Constr.*, vol. 75, pp. 22–32, 2017.