Semantic Web 0 (0) 1
IOS Press

BOT: the Building Topology Ontology of the W3C Linked Building Data Group

Mads Holten Rasmussen [a,*], Maxime Lefrançois [b], Georg Ferdinand Schneider [c] and Pieter Pauwels [d]

[a] *Department of Civil Engineering, Technical University of Denmark, Denmark*
E-mail: mhoras@byg.dtu.dk
[b] *Mines Saint-Étienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516, LHC, Institut Henri Fayol, F - 42023 Saint-Étienne France*
E-mail: maxime.lefrancois@emse.fr
[c] *Fraunhofer Institute for Building Physics IBP and Technische Hochschule Nürnberg, Nürnberg, Germany*
E-mail: georg.schneider@ibp.fraunhofer.de
[d] *Department of the Built Environment, Eindhoven University of Technology, Netherlands*
E-mail: p.pauwels@tue.nl

Editor: Krzysztof Janowicz, University of California, Santa Barbara, USA
Solicited reviews: Tomi Kauppinen, Aalto University, Finland; Yingjie Hu, University at Buffalo, USA; Simon Scheider, Utrecht University, Netherlands

Abstract. Actors in the Architecture, Engineering, Construction, Owner and Operation (AECOO) industry traditionally exchange building models as files. The Building Information Modelling (BIM) methodology advocates the seamless exchange of all information between related stakeholders using digital technologies. The ultimate evolution of the methodology, BIM Maturity Level 3, envisions interoperable, distributed, web-based, interdisciplinary information exchange among stakeholders across the life-cycle of buildings. The World Wide Web Consortium Linked Building Data Community Group (W3C LBD-CG) hypothesises that the Linked Data models and best practices can be leveraged to achieve this vision in modern web-based applications. In this paper, we introduce the Building Topology Ontology (BOT) as a core vocabulary to this approach. It provides a high-level description of the topology of buildings including storeys and spaces, the building elements they contain, and their web-friendly 3D models. We describe how existing applications produce and consume datasets combining BOT with other ontologies that describe product catalogues, sensor observations, or Internet of Things (IoT) devices effectively implementing BIM Maturity Level 3. We evaluate our approach by exporting and querying three real-life large building models.

Keywords: Linked Data, Building Information Modelling, Ontologies, Building Topology Ontology

1. Introduction

The global Architecture, Engineering, Construction, Owner and Operation (AECOO) industry contributes significantly to the economy of industrialised and emerging countries (e.g. 2.5M employees [21] and 15.9% of the gross value added in Germany [20]). The specific characteristics of the industry make it challenging to successfully handle projects in this domain. One challenging characteristic is the fragmented structure of the industry, as it is composed of numerous small and

*Corresponding author. E-mail: mhoras@byg.dtu.dk.

medium-sized companies. In addition, interdisciplinary stakeholders from different trades each using own special software tools [6] need to work together and exchange information over the whole life cycle of a project [62]. Current approaches rely on the establishment of a temporary project organisation for each new project. Therefore, it is challenging to carry the gathered project information and best practices onwards to the next project as stakeholders change.

During the whole life cycle of a building, vast amounts of data are generated, exchanged and processed. The facilitation of a seamless exchange of project information over the whole life cycle of a construction facility as well as between multiple, interdisciplinary stakeholders is a fundamental necessity for the successful accomplishment of these projects. Due to the fragmented structure of the industry, this information supply chain is often reestablished from near scratch with each new project organisation, resulting in new custom data structures for every project, represented in individual ever-changing unstructured spreadsheets and documents.

Building Information Modelling (BIM) is a methodology under research since decades [18], which advocates the seamless exchange of all information between related stakeholders by the use of digital technologies. It allows addressing the above-described problems in the information exchange in AECOO projects. A growing interest can be found in the BIM method, even for existing buildings [68], and its adoption gains momentum as, similar to other industries, the AECOO industry experiences a ubiquitous introduction of Information and Communication Technologies (ICT) in the course of the digital transformation of the domain. By now, BIM as a method has established itself globally in the construction industry, making the industry shift significantly towards full digitisation. Yet, it still suffers from the diversity of (custom) data structures and use of unstructured data in documents (BIM Level 2).

This shift towards the use of BIM happens according to a number of maturity levels. Figure 1 depicts the four maturity levels that are defined by Bew and Richards [7] for the BIM methodology. These levels indicate how maturely the BIM methodology is implemented in a given company, and each level outlines the technological requirements for its successful realisation at that level. These levels serve as a guideline for the evolution steps of the adoption of BIM by industry and policy makers [7]. Maturity level 0 is the "Pre-BIM"-phase, where building information is exchanged in an uncoordinated manner based on drawings (CAD drawing and paper-based exchange). In Level 1, companies and stakeholders collaborate in a file-based manner, and focus mostly on 2D and 3D geometric modelling; whereas companies in Level 2 work with full BIM models, which are typically understood to be complex 3D models enriched with big amounts of information (material data, usage data, design constraints, etc.). Collaboration in Level 2 is mainly file-based still. When achieving the highest maturity Level 3, it is envisioned that process and information is exchanged purely on a web-scale and fully integrated over disciplines and companies. BIM Level 3 can be compared to BIM Level 2, similar to how the Web of Data can be compared the Web of Documents.

Currently, the AECOO industry is situated at Level 0, 1, or 2 of this diagram, depending on the region in the world, where (manual) file-based information exchange is still the state of the art. Exchange approaches rely on files, e.g. Industry Foundation Classes (IFC) [33], and file containers, e.g. ISO 19650-1 [34], or Common Data Environments (CDEs) for the centralised web-based storage and exchange of construction-related files (e.g. Autodesk A360,[1] Microsoft 365[2]). The use of a CDE is also stipulated in European BIM implementation guidelines [65], however, a common flaw of these approaches is that through the distribution of information across files the linking of information at the data level, as required for BIM Maturity Level 3, is not possible. Also tracking of changes is only possible at the file level, which is a major limitation [57].

In essence, BIM Maturity Level 3 is, apart from high-level descriptions [8], rather undefined, and approaches for implementation are missing (see Section 2). However, it is clear that, for BIM Maturity Level 3, information is exchanged on the Web using open standards, and interoperable and decentralised model servers allow collaborative work on interoperable models and structured data. From this assumption, one may define the

[1] https://a360.autodesk.com/
[2] https://www.microsoft.com/microsoft-365/

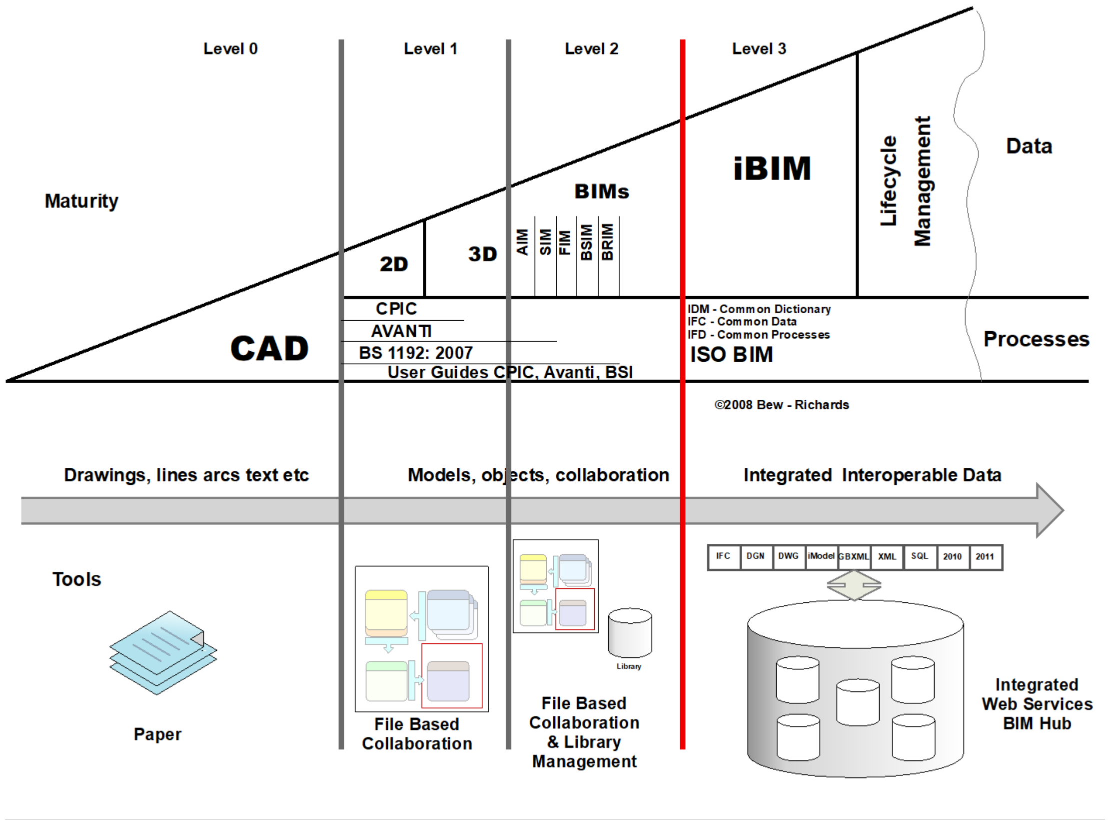

Fig. 1. BIM Levels of Maturity, with the web-based BIM Level 3 on the far right (copyrighted image: [7]).

following general requirements for BIM Maturity Level 3:

REQ1 Support of web-based information exchange [8];
REQ2 Use of an information hub to allow collaborative, web-based workflows among interdisciplinary stakeholders [8];
REQ3 Use of a set of interoperable, flexible, and open, standards covering different domains;
REQ4 Support of distributed data integration, linking and tracking at data level.

The vision of the Linked Building Data (LBD) Community Group (CG)[3] of the World Wide Web Consortium (W3C) is that adopting Linked Data and Semantic Web Technologies [17] in the AECOO industry would help covering these requirements, therefore following the same evolution as experienced in the World Wide Web by moving from a web of documents to a web of data [9, 47].

In this paper, we report on a collaborative effort led in the context of the LBD CG to develop a lightweight [16] and extensible ontology[4] named the Building Topology Ontology (BOT), which provides a high-level description of the topology of buildings including storeys and spaces, the building elements they may contain, and the 3D mesh geometry of these spaces and elements. Precursors of the ontology have been published in earlier publications of the authors [53, 54]. Since then, the ontology has been substantially revised and substantial changes have been applied by the active development through members of the W3C LBD CG group. Since its initial (v0.1.0, [53]) and intermediary (v0.2.0, [54]) version, the ontology has grown from four to seven classes and from 5 to 14 object properties in its most recent release (v0.3.1) documented in this paper. In particular, the relationship to geometrical data has been

[3] https://www.w3.org/community/lbd/

[4] An ontology is a formal, explicit specification of a shared conceptualisation of a domain [26].

added as described in Section 3. In addition, multilingual labels and descriptions have been added to the concepts and relationships of the ontology. BOT is lightweight and intended to be used in combination with other ontologies (e.g. to represent product information, sensor observations, Internet of Things (IoT) devices, complex geometry, or project management data), to provide a simple option to reach semantic interoperability and enable data integration on the web by the AECOO industry.

The rest of this article is organised as follows. In Section 2 we provide an analysis of the current state of the art in moving the AECOO industry in the direction of the Web of data. Then Section 3 details the most recent version of the BOT ontology, and the proposed conceptual alignment to the DOLCE Ultralite ontology [23], and to other related ontologies. Section 4 describes how BOT is expected to be used in combination with other ontologies. It also reports on existing applications that produce or consume BOT datasets. Section 5 provides an evaluation of the export and query of BOT datasets for three large building models.

2. State of the Art

Industry practitioners actively work towards BIM Maturity Level 3, and, as a result, different open source community-based software projects have evolved in the past years to fulfil requirements REQ1 and REQ2 (e.g. Flux.io,[5] vA3C[6] and speckle.works[7]). These are aiming at enabling direct information exchanges, mainly concerning geometry, between native Computer Aided Design (CAD) and BIM software using web Application Programming Interfaces (APIs).

In terms of information exchanges, the standard schema for the exchange of BIM data is the Industry Foundation Classes (IFC) [33], which is a data model described in EXPRESS [31] and which has a strong focus on the representation of 3D geometry [49]. However, IFC does not fulfil requirements REQ3 and REQ4 in that it is not web-compliant, and fails at enabling the integration of building data with other types of data on the Web. Arguably, a better move to bridge this gap is to adopt the Linked Data principles [5] including the use of semantic web standards and technologies [17] such as the Resource Description Framework (RDF) [42], the Web Ontology Language (OWL) [30], and the SPARQL Protocol and RDF Query Language (SPARQL) [29]. Therefore, various works investigated how Semantic Web technologies can be used for the AECOO industry. We overview these works in the rest of this section, using as a starting point a recent survey by Pauwels et al. [47]. We hereby also briefly indicate how geospatial data standards fit obtaining BIM Level 3 using semantic web technologies.

2.1. *IFC in OWL and OWL in IFC*

A pioneer initiative aiming at integrating IFC and OWL was named ifcOWL and proposed in 2005 and 2009 by Beetz et al. [2, 3].

2.1.1. *The ifcOWL Ontology and simplification initiatives*

Heavily relying on this early work, Pauwels and Terkaj [46] implemented a direct mapping of the EXPRESS schema to OWL, and applied this transformation to the IFC EXPRESS schema to produce the ifcOWL ontology.[8] In doing so, a number of criteria was followed, the most important one being that the resulting ontology was required to be fully backwards compatible with the EXPRESS schema of IFC. As a result, ifcOWL has two major drawbacks.

A) Complex structure of ifcOWL The proposed systematic transposition results in modelling choices that are inconsistent with the best practices in the Semantic Web domain (e.g., defining a class for booleans or relations). Also, the resulting ifcOWL includes many syntactical constructs stemming from the EXPRESS source schema (e.g. ordered lists, objectified relations, objectified properties, 'select' classes, and 'enumeration' individuals). Even though this enables round-tripping between IFC documents and ifcOWL ontologies, it makes ifcOWL, like IFC itself, too complex, hard to manage, hard to understand, and also makes reasoning highly inefficient [49, 63].

[5] Discontinued, no longer online
[6] https://va3c.github.io/
[7] https://speckle.works/
[8] https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#

B) Size of ifcOWL ifcOWL contains in a single ontology all the terms of the IFC specification, including terms related to lists, datatypes, time scheduling, cost estimation, or quantitative units. This size of ifcOWL hampers its understanding and usability by developers that may need just a few concepts. In other words, the highly needed *modularity* and *extensibility* are entirely missing. For example, the latest version of ifcOWL for IFC4_ADD2 consists of 1331 classes and 1599 properties. Ongoing work aims at extending IFC towards roads [36] and bridges [69], which will ultimately make the resulting ifcOWL even bigger. However, Terkaj and Pauwels [64] have later suggested an approach to generate a modular version of ifcOWL, based on the modules that are present at the core of IFC.

Aiming to resolve the above drawbacks, a number of efforts then aimed at defining mechanisms to automatically simplify building models described with the ifcOWL ontology (which can be referred to as ifcOWL datasets). IFC Web of Data (IFC-WoD) [43] and SimpleBIM [45] both cut away elements like geometric data and intermediate EXPRESS-derived relation instances between objects. These approaches have been proven successful, yet they are amendments to an ontology that is intrinsically insufficient because of its backlog (the IFC EXPRESS schema). Indeed, the result remains relatively close to the EXPRESS version of IFC, instead of aiming first at best practices and publishing modular ontologies that are based on known and proven ontology design patterns.

In terms of simplification for ifcOWL, BimSPARQL [70] uses another approach that leverages the application of SPARQL Inferencing Notation (SPIN) rules to provide shortcuts, thereby making it simpler to query an ifcOWL dataset. This allows for bypassing the intermediate node between a space and its contained elements, for example. The work also demonstrates rules that perform geometric operations on geometry, and in general, it showcases a promising approach for data extraction from BIM models. The size-related drawbacks of ifcOWL are, however, still persistent with this approach, and a new semantic web-born set of ontologies is needed for this industry.

2.1.2. Alternative approaches

Alternative approaches aimed to make building data available over the web in a more structured format, typically also deploying semantic web technologies.

Metadata in IFC files Beetz et al. [4] proposed to use existing features of the IFC model to allow for the direct incorporation of meta-data in the IFC document that give access to external RDF data. In this approach, the core of IFC, and in particular the geometry, can still be used, while also allowing to link to external RDF data. This approach addressed the extensibility issues of IFC, while avoiding to abandon the EXPRESS schema for IFC. Although the resulting IFC documents are compatible with IFC, they still centralise all the information. Therefore, BIM Level 3 requirements REQ2 and REQ4 could not be covered. At best, this presents a transitional approach towards the implementation of BIM Level 3.

Annotation of online resources with IFC concepts Gao et al. [24] defined a domain ontology of IFC, with the goal to annotate online resources with the IFC data model, and thus use IFC in combination with semantic web technologies to perform information retrieval (IFC-IR) [25, 40]. They demonstrate with their approach that IFC data on the Web can efficiently be retrieved using SPARQL queries. However, this approach does not fulfil BIM Level 3 requirement REQ3, as the file-based exchange mechanism still prevails.

BIMSO/BIMDO The foundation ontology BIM Shared Ontology (BIMSO) has been defined for the AECOO industry, with the purpose of being extended with various building domain ontologies [44]. The authors claim that the ontology only contains a few classes and relationships scoped at describing a building's elements, levels, spaces and construction phases, and relies on the full Uniformat II classification system for further organising the elements. A separate ontology, the BIM Design Ontology (BIMDO), provides the necessary object properties to describe relationships between elements, subdivision of zones and to quantify these relationships [44]. However, these ontologies have not been made publicly available, which violates the first principle of the Linked Data deployment scheme.

2.2. The W3C LBD CG

Many other ontologies have been developed for the AECOO domain, subsets of it, or related domains such as sensors and actuators, or the IoT. This consistently leads to contradictory redefinition of common terms [53] such as "building", which, as of April 2019, is defined in 690 separate ontologies in the Linked Open Vocabulary [66] [9]. The most related ontologies include DogOnt [12], BIMSO [44], the Smart Appliances REFerence Ontology (SAREF) ontology and its extension for buildings SAREF4BLDG [15, 67], ThinkHome [59], Smart Energy Aware Systems (SEAS) [37, 39], Brick [1].

The W3C LBD CG was created to bring together experts in the area of BIM and Web of Data technologies. One of its goals was to identify and align existing initiatives to model building data across the life cycle of buildings. The alignment between the terms in these ontologies was studied[10] [60, 61]. Finally, a proposal was made to decouple the description of building data according to different complementary aspects, including the topology of buildings, geometry, building-related properties (e.g., room temperature, wall thickness, wall thermal conductivity), building-related products (doors, windows, beams, ducts, pipes), project management, management of properties.

Part of the data in these categories is not specific to buildings and may be described using existing standardised vocabularies, according to the best practices [9]. For example: (1) the Semantic Sensor Network Ontology (SOSA)/Semantic Sensor Network Ontology (SSN) ontology [28] can be used to describe observations and actuations of properties in buildings, (2) schema.org can be used to describe products, (3) SAREF can be used to describe IoT devices [15].

When no existing ontology could be reused, ontology proposals were made. For example, the Ontology for Property Management (OPM) [57] can be used to describe property states, thereby allowing property values to evolve over time while keeping track of their history. It extends the SEAS ontology [37, 39] and the Provenance Ontology (PROV-O) [35].

Finally, it has been decided that the group was legitimate to develop a lightweight ontology providing a high-level description of the topology of buildings including storeys and spaces, the building elements they may contain, and the geometry of these spaces and elements. The rest of this article describes the result of this development, the BOT ontology, which is currently the most mature report of the W3C LBD CG [58].

The group aimed at creating a lightweight BOT ontology that would not have the same drawbacks found in IFC in terms of size and complexity. Reuse of existing ontologies was an important priority, which includes ontologies for specialised areas, as mentioned above, such as sensor data, product data, geometry, and so forth. Such detailed ontologies are not to be incorporated in BOT, yet, they are meant to be linked to whenever BOT-compliant RDF data is produced (see further on in this article). As an example, the geospatial domain is a very important reference domain for the AECOO industry. Instead of including the geospatial domain within the scope of BOT, the group aimed to limit to referential topological concepts of a building, which can then reference geospatial data that is represented using its own standards (e.g. CityGML).

3. The Building Topology Ontology (BOT)

The scope of BOT is to explicitly define necessary relationships between the sub-components of a building. As such, it aims to provide the means for representing interlinked information in a future (semantic) web driven AECOO industry, satisfying the recommendation of reusing terms already described in well-known vocabularies wherever possible [9].

The first version of BOT was presented in [53] and an increment in [54]. Since then, the ontology has been further extended to accommodate modelling issues raised by the community. This section first overviews the competency questions of the ontology, then provides an overview of the current version v0.3.1 of BOT, then details its main components, and finally discusses the alignments with related ontologies.

3.1. Overview of the BOT Ontology Competency Questions

The Competency Questions (CQs) for BOT were raised by the community during the W3C

[9] https://lov.linkeddata.es/dataset/lov/terms?q=building

[10] https://docs.google.com/document/d/1wSxpE5O6jntcIuhey7Uv0o0ZAU1Dz-ZSICuuxbwGvCA#

LBD CG group community calls, on the public mailing list, during the Linked Data in Architecture and Construction (LDAC) workshop series, and on the project repository on GitHub.[11] They are listed on the documentation website of BOT https://w3id.org/bot#, and copied below.

- **CQ1** What are the zonal constituents of the overall building (e.g. site, building, storey, space)?
- **CQ2** What smaller zones are contained inside the larger zone (e.g. space zone contained in the storey zone; contained in the building zone; contained in the site zone)?
- **CQ3** What zone(s) are adjacent to or intersecting with a zone?
- **CQ4** What are the tangible building elements that the building consists of and what are the sub elements of these building elements?
- **CQ5** Which element(s) are contained inside the 3D-extent of a particular zone? Which elements are adjacent to the zone? Which elements are intersecting with the zone?
- **CQ6** How to assign metadata to a connection between zones, elements or zones and elements?
- **CQ7** What is the 3D model(s) (including geometry, material, etc.) of a zone/ element?

The difference between *zone* and *element* is common in the building and construction domain. An element is a concrete and tangible object whereas a zone is typically just air encapsulated by elements. In construction projects, spaces and zones are the physical frames for some functional requirements of the client (e.g. there is a need for a space that can facilitate two office workers with each their desk and with these requirements for the indoor climate). It is common practice to use these zones as placeholders for functional requirements even before they exist in the designed or the actual building. The functional requirements of the zones are translated by the designers into boundary conditions to technically equip these zones, which results in a number of physical building elements (e.g. number of ventilation terminals, work stations, lighting fixtures, hospital beds etc. and the specifications of these). It is therefore fundamental for anyone from the target audience working in the construction and related industry to have these concepts.

[11] https://github.com/w3c-lbd-cg/bot/issues

3.2. *Overview of the BOT Ontology*

The version v0.3.1 of BOT described in this paper consists of 7 classes, 14 object properties, and one datatype property, with a Description Logics (DL) expressivity of $\mathcal{SRI}(\mathcal{D})$. BOT is in the OWL 2 RL profile [30, Sec. 10.3]. It is documented and available at its Uniform Resource Identifier (URI) https://w3id.org/bot following the recommended best practices. Changes across the versions of BOT are tracked and listed in the documentation,[12] and in the history of the repository.[13] Terms defined in the BOT ontology are identified by URIs in the namespace https://w3id.org/bot#, which we shorten in the rest of this article with the prefix bot:, (registered at http://prefix.cc) as listed below.

```
@prefix bot: <https://w3id.org/bot#> .
```

The high level terminology of the ontology is illustrated in Figure 2. BOT has three main classes: bot:Zone, bot:Element, and bot:Interface required for CQs **CQ1,4,6**. A bot:Zone is a part of the world that has a 3D spatial extent (i.e., building, space, thermal zone, fire cell) or a sub-part or an aggregation of such parts. A bot:Element is a constituent of a construction entity with a characteristic technical function, form or position [32, Section 3.4.7]. It can be any tangible object (product, device, construction element, etc.) that exists in the context of a zone, i.e., a part of the world. A bot:Interface is a part of the world that is common to some specific zones and elements, and at the boundary of at least one of them.

As illustrated in Figure 3 and required to cover **CQ1**, four sub-classes of bot:Zone are defined: bot:Site, bot:Building, bot:Storey, and bot:Space. Also, three sub-properties of bot:hasElement are defined to cover **CQ5**: bot:containsElement, bot:adjacentElement, and bot:intersectingElement. Finally, one may assign a 3D model to any bot:Zone or bot:Element, either object property bot:has3DModel or datatype property bot:hasSimple3DModel. This covers **CQ7**.

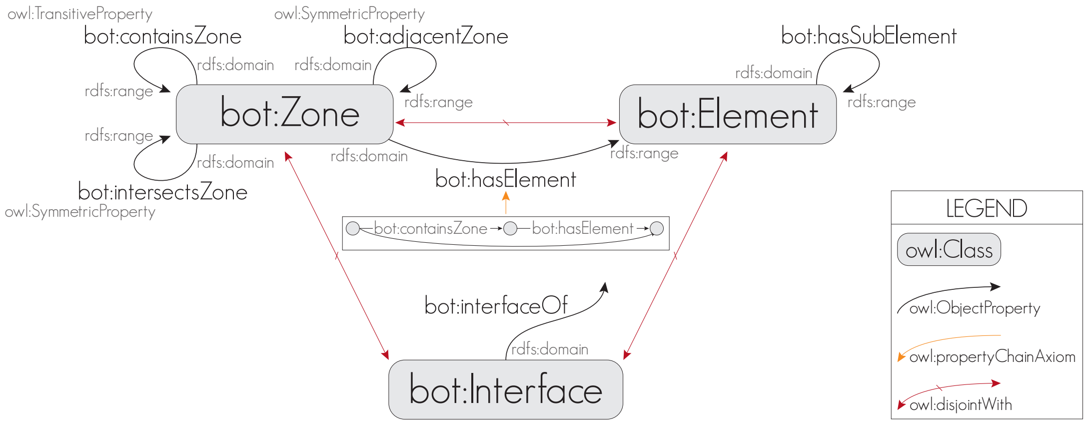

Fig. 2. Illustration of the main three classes of BOT, which are pairwise disjoint, and the main properties used to link instances of these classes. The domain, range, and potentially transitive or symmetric aspect of object properties is illustrated. Objects of the bot:interfaceOf property typically are instances of bot:Zone or bot:Element. The property chain bot:containsZone ∘ bot:hasElement is a sub-property of the property bot:hasElement.

3.3. Zones and sub-Zones

A bot:Zone is defined as a *part of the world that has a 3D spatial extent.*[14] Four sub-classes of bot:Zone are defined: bot:Site, bot:Building, bot:Storey and bot:Space. The concept of bot:Zone may be reused to describe moving habitable structures, such as trains or boats, or virtual buildings, such as in virtual reality software. Three topological relationships are defined between zones:

bot:containsZone is transitive, and links a zone to another one it fully contains. Three sub-properties of bot:containsZone are defined: bot:hasBuilding, bot:hasStorey and bot:hasSpace, whose ranges are bot:Building, bot:Storey and bot:Space, respectively. These properties can be used to group or subdivide zones as illustrated in Figure 4, and cover **CQ1,2**;

bot:adjacentZone is symmetric, and links two zones that share part of their boundary (in the topological sense);

[12] https://w3id.org/bot/#changes

[13] https://github.com/w3c-lbd-cg/bot/commits/master

[14] This definition is inspired by the definition of Spatial Thing in the DOLCE Ultralite ontology [23].

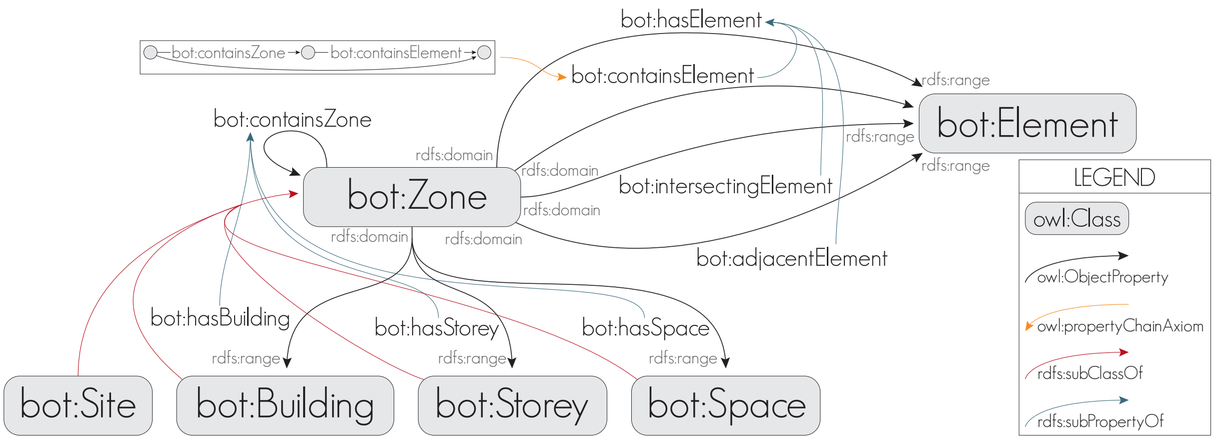

Fig. 3. Illustration of the four sub-classes of bot:Zone and the three sub-properties of bot:hasElement. The domain and range of object properties is illustrated. The property chain bot:containsZone ∘ bot:containsElement is a sub-property of the property bot:containsElement

bot:intersectsZone is symmetric, and links two zones whose 3D spatial extent is partly shared (e.g. a stair well intersecting several storeys).

bot:adjacentZone and bot:intersectsZone together cover **CQ3**. Other more detailed calculi to define topological relationships among regions exist, such as the Region Connection Calculus (RCC) [51]. However, to keep BOT as simple as possible we only consider bot:containsZone, (unification of tangential proper part and non-tangential proper part), bot:adjacentZone (equivalent to externally connected and bot:intersectsZone, (a domain specific generalisation of externally connected). Also, different to RCC, the BOT topological relations link different conceptual entities (zones and zones, zones and elements).

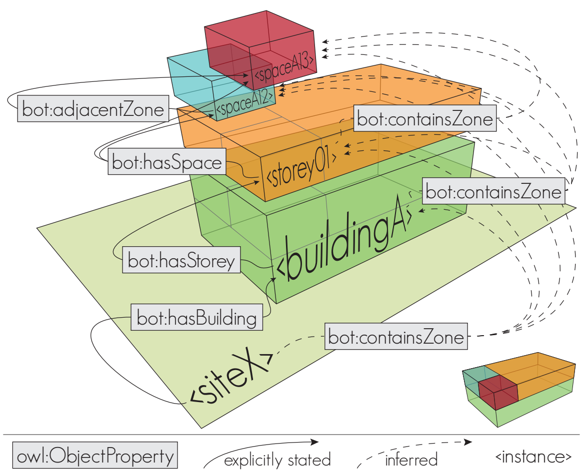

Fig. 4. Zones in BOT follow a Matryoshka doll principle where one zone can be contained within another zone and so forth [54].

The classes of BOT can be used not only for existing buildings but can also be used to create requirements of a future building. For example, Rasmussen et al. [56] defines the client's requirements for spaces of a future building as sub-classes of bot:Space.

3.4. Elements and sub-Elements

A bot:Element is defined as a *constituent of a construction entity with a characteristic technical function, form or position [32, Section 3.4.7]*. Elements can *host* sub-elements, which is defined using the bot:hasSubElement property. This covers **CQ1,5**. For example a window may have an outdoor temperature sensor as a sub-element and an air handling unit has at least one fan as a sub-element.

Three main topological relationships between zones and elements are defined, so as to cover **CQ5**:

bot:adjacentElement links a zone to an element that shares part of its boundary;

bot:intersectingElement links a zone to an element whose 3D extents is partly shared;

bot:containsElement links a zone to an element it contains.

The latter property is used in a property chain axiom that formalises the fact that: *if a zone contains a zone that contains an element, then it contains that element*:

$$\text{bot:containsZone} \circ \text{bot:containsElement} \sqsubseteq \text{bot:containsElement}$$

A super-property of these three properties, bot:hasElement, is defined to indicate a generic relationship between a bot:Zone and a bot:Element. The intended use of this relationship is not to be stated explicitly, but to be inferred from its sub-properties. It allows, for example, to query for all the doors of a building given that they have an adjacency to spaces contained in the building. Property bot:hasElement is also used in a property chain axiom that formalises the fact that: *if a zone contains a zone that has an element, then it has that element*:

$$\text{bot:containsZone} \circ \text{bot:hasElement} \sqsubseteq \text{bot:hasElement}$$

3.5. Interfaces

The class bot:Interface is used to describe the relationship between some specific zones and elements in detail, and covers **CQ6**. This class can be used to qualify (i.e., attach additional information to) any of the aforementioned topological relationships between zones, elements, or zones and elements. Figure 5 illustrates two interfaces between two zones and a wall. The concept of bot:Interface is useful in different situations:

a) the heat transmission area of the surface between a space and an adjacent wall can be used to determine the heat loss from that space through this wall;
b) the localisation of the intersection between a pipe and a wall can be used to specify where to apply fire sealing;
c) the type of access between two zones can be used to specify access restrictions for use in indoor navigation.

An interface is assigned to elements or zones using the bot:interfaceOf property. The domain of that bot:interfaceOf is bot:Interface. Objects of the bot:interfaceOf property typically are instances of bot:Zone or bot:Element.

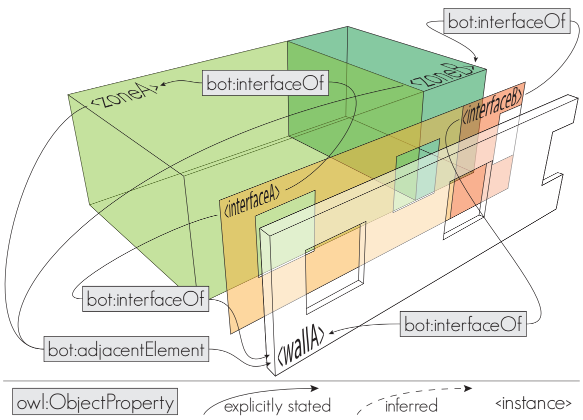

Fig. 5. Two interfaces between two zones and a wall. Interfaces can be used to qualify (i.e., attach additional information to) topological relationships between zones, elements, or zones and elements [54].

3.6. Assigning Geometry

The last CQ **CQ7** requires BOT to provide a simple means to link a zone or element to its 3D model. How the model is encoded is not in the scope of BOT, but the documentation provides some examples.

Any bot:Zone or bot:Element can be assigned a 3D Model (including geometry, material, etc.), using some existing data format for 3D models. Two properties are defined for this:

Datatype property bot:hasSimple3DModel can be used if the 3D Model can be encoded as a literal. We encourage the use of URIs for mediatype descriptions with the IANA authority.[15] For example https://www.iana.org/assignments/media-types/model/3mf for the mediatype `model/3mf`. Other mediatypes for Wavefront OBJ [22], STP, IFC, W3D, etc. can be defined. If the data format is textual, then the lexical form of the 3D Model literal should be encoded as a Unicode string. For binary data formats, the lexical form of the literal should be its base32 encoding.

Object property bot:has3DModel can be used to link a bot:Zone or bot:Element to some URI that identifies a 3D Model. This 3D Model can then be described using some dedicated RDF vocabulary. Else, the 3D Model URI could be dereferenceable, and when looking up the URI one could retrieve a representation of the 3D Model with some existing data format for 3D models.

Bonsma et al. [13] discusses different considerations for describing complex geometry with ontologies, including references to the ontoBREP approach [50] and the ifcOWL approach [48]. Then the 3D model geometry, which is specified relative to the local coordinate system of the model, can be positioned in a global Geospatial Information Systems (GIS) context using the zero point of the site.

Figure 6 is a screenshot of a demonstration web-based software that renders a zone and its adjacent element instances in the browser. The 3D geometry of these zones and elements is a simple mesh geometry described using OBJ literal that is automatically extracted from a BIM authoring tool. This demonstration illustrates how existing web frameworks and libraries can be used out of the box to implement powerful solutions based on BOT, which may be used by users in the AECOO industry across the building lifecycle (see also Section 4). This demo implements functionalities that combine Linked Data and geometry.

3.7. Alignment to other ontologies

BOT is designed to function as a central element in the interdisciplinary communication of the AECOO sector. In addition, it aims at being the

[15] IANA is the Authority responsible for registering mediatypes, among other.
[16] https://madsholten.github.io/BOT-Duplex-house

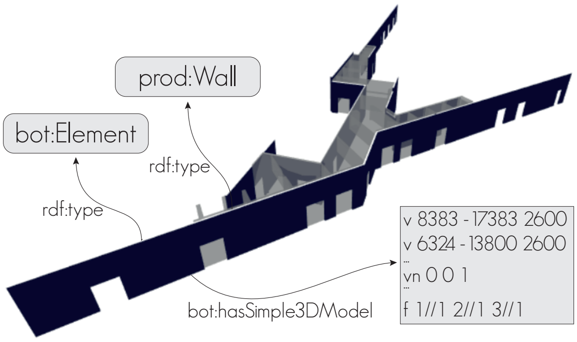

Fig. 6. Example of a graphical feedback from a request for all wall elements adjacent to a particular space using BOT terminology. The 3D model is described as an OBJ-formatted [22] mesh. A simple demo can be found online[16].

key entry point to connect AECOO sector to adjacent domains. Moreover, alignments potentially allow to define automatic converters from datasets described with one ontology to another.

As there are numerous ontologies available in the AECOO domain we only describe two alignments in this paper: (1) the alignment to ifcOWL [46] a well accepted standard in the construction industry; and (2) to the DOLCE Ultralite upper ontology (DUL) [23], which is a foundational ontology meant to support broad semantic interoperability among domain-specific ontologies by providing a common starting point for the formulation of definitions.

Alignment to ifcOWL As a number of ontologies already exist in the construction domain, alignments of BOT to six commonly used domain ontologies are defined in [60, 61]. The formal alignments are provided as separated ontologies[17] Other formats could be also possible, e.g. Alignment Format [19]. One of these alignments is between BOT and ifcOWL. The concepts ifc:IfcSite, ifc:IfcBuilding, ifc:IfcBuildingStorey and ifc:IfcSpace can be straightforwardly specialised from their respective BOT concepts, i.e. bot:Site, bot:Building, bot:Storey, bot:Space. This also applies to the description of tangible building elements, i.e. specialising ifc:IfcElement from bot:Element. As ifcOWL uses classification to describe relationships among concepts, e.g. ifc:IfcRelAggregates and ifc:IfcRelDecomposes, no correspondences to object properties of BOT are defined [60].

Alignment to the DOLCE Ultralite ontology In addition to domain specific extensions, this work presents correspondences to upper ontologies such as DUL [23]. The concept bot:Zone and bot:Interface are specialised from dul:PhysicalObject, which is the concept in DUL of objects that are spatially located and have their proper space region. bot:Site is specialised from dul:PhysicalPlace meaning its location is inherent. bot:Building, bot:Storey, bot:Space and bot:Element, are specialised from dul:DesignedArtifact, which are physical artefacts described by a design. The object property bot:has3DModel is aligned to dul:hasRegion, and its range is further specialised to dul:SpaceRegion, which is the dimensional space that is used to localise the bot:Zone or bot:Element. Among object properties the following correspondences are defined:

– bot:containsZone and bot:containsElement are specialised from dul:hasPart;
– bot:adjacentZone and bot:adjacentElement are specialised from dul:hasCommonBoundary;
– bot:intersectsZone, bot:intersectingElement, and bot:interfaceOf are specialised from dul:overlaps.

4. Using BOT in practice

In this section, we overview how the BOT ontology can be used in combination with other ontologies.

4.1. Sub-typing BOT classes and properties

An external ontology can directly extend BOT defining sub-classes of BOT classes. Figure 7 illustrates one approach where the class fso:Heater from a fictive Flow Systems Ontology (FSO) is defined as a sub-class of bot:Element. From the explicit axioms illustrated with plain arrows in this knowledge base, a DL reasoner can infer that if inst:heater33 is of type fso:Heater, then it is also of type bot:Element, thereby giving it a more generic abstraction understandable by other domains.

BOT can also be extended with more specific properties. Figure 8 illustrates an approach where a new property fso:heatedBy is defined as a

[17] https://w3c-lbd-cg.github.io/bot/#AlignmentModules.

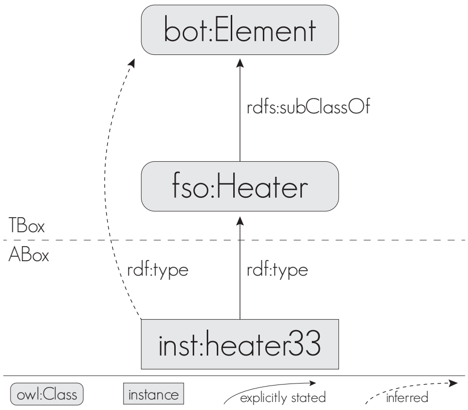

Fig. 7. Linking by defining sub-classes.

sub-property of bot:containsElement, and having range fso:Heater. From the explicit axioms illustrated with plain arrows in this knowledge base, a DL reasoner can infer that inst:spaceA2 contains inst:heater33, and that this element is of type fso:Heater.

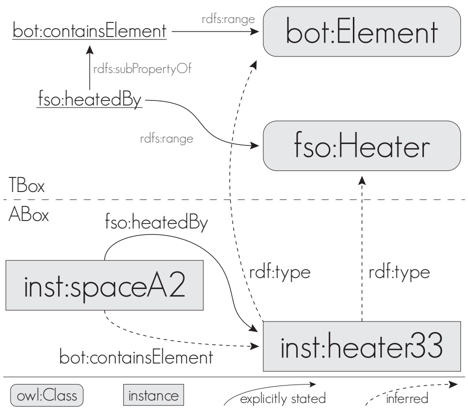

Fig. 8. Linking by defining subClasses.

4.2. *Catalogues of products*

An external ontology could define a catalogue of products including windows, walls, ducts or defibrillators. An instance of one of these classes can also be an instance of bot:Element. This can be explicitly asserted, or inferred from topological relations with other instances of bot:Zone or bot:Element. Figure 9 illustrates a knowledge base where an individual inst:prodABC is asserted to be an instance of the class product:Defibrillator, and to be contained in the zone inst:spaceA2. The dashed arrows illustrate the relationships that can be automatically inferred using DL reasoning.

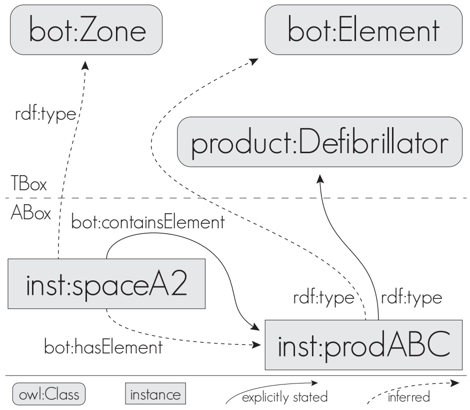

Fig. 9. Example of an instance of both (1) a class defined in a hypothetical ontology of products, and (2) the bot:Element class. In this example relations illustrated using plain arrows are explicit, and relations illustrated using dashed arrows can be automatically inferred using DL reasoning.

4.3. *Quantifying the properties of Zones, Elements, and Interfaces*

Different approaches for assigning values to the properties of some bot:Zone, bot:Element, or bot:Interface were discussed by Rasmussen et al. [57]. Assume one wants to assert that the input and output temperatures of a pipe are currently 61.0°C and 42.0°C, but the requested output temperature of that pipe is 50.0°C.

The most simplistic form (L1 in [57]) consists in directly linking the pipe to each of its temperature values, described as literals or as individuals. For example, the snippet below defines the three temperatures using the Custom Datatypes (CDT) Unified Code for Units of Measure (UCUM) datatype [38].

```
@prefix cdt:
    <http://w3id.org/lindt/custom_datatypes#>.

ex:hasCurrentInputTemp a owl:DatatypeProperty .
ex:hasCurrentOutputTemp a owl:DatatypeProperty .
ex:hasRequestedOutputTemp a owl:DatatypeProperty .

<pipe1> a bot:Space ;
  ex:hasCurrentInputTemp "61.0 Cel"^^cdt:ucum ;
  ex:hasCurrentOutputTemp "42.0 Cel"^^cdt:ucum ;
  ex:hasRequestedOutputTemp "50.0 Cel"^^cdt:ucum .
```

The snippet below represents the same knowledge but using the QUDT ontology [27].

```
@prefix qudt11: <http://qudt.org/1.1/schema/qudt#> .
@prefix qudtu11: <http://qudt.org/1.1/vocab/unit#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

ex:hasCurrentInputTemp a owl:ObjectProperty .
ex:hasCurrentOutputTemp a owl:ObjectProperty .
ex:hasRequestedOutputTemp a owl:ObjectProperty .

<pipe1> a bot:Element ;
  ex:hasCurrentInputTemp _:qv_ci ;
  ex:hasCurrentOutputTemp _:qv_co ;
  ex:hasRequestedOutputTemp _:qv_ro .

_:qv_ci a qudt11:QuantityValue ;
  qudt11:unit qudu11:DegreeCelsius ;
  qudt11:numericValue "61.0"^^xsd:double .

_:qv_co a qudt11:QuantityValue ;
  qudt11:unit qudtu11:DegreeCelsius ;
  qudt11:numericValue "42.0"^^xsd:double .

_:qv_ro a qudt11:QuantityValue ;
  qudt11:unit qudtu11:DegreeCelsius ;
  qudt11:numericValue "50.0"^^xsd:double .
```

These approaches cannot describe the context in which the value assignment holds. It is not explicit that there are two different values for the same property and another value for another property.

A more flexible approach, relying on specific properties as described in the SOSA/SSN standard [28], consists in using ex:Temperature as a class, and associating two different instances of that class to the pipe (the input and output temperature) using different properties (ex:hasInputTemperature and ex:hasOutputTemperature). The snippet below illustrates this approach using SOSA/SSN, SEAS [37], and the CDT UCUM datatype.

```
@prefix sosa: <http://www.w3.org/ns/sosa/>.
@prefix cdt:
    <http://w3id.org/lindt/custom_datatypes#>.

ex:Temperature a owl:Class ;
  rdfs:subClassOf sosa:ObservableProperty .

ex:hasInputTemperature a owl:ObjectProperty .
ex:hasOutputTemperature a owl:ObjectProperty .

seas:ComfortEvaluation a owl:Class .
sosa:Observation a owl:Class .

<pipe1> a bot:Space ;
  ex:hasInputTemperature <pipe1#input> ;
  ex:hasOutputTemperature <pipe1#output> .

<ci> a sosa:Observation ;
  sosa:observedProperty <pipe1#input> ;
  sosa:hasSimpleResult "61.0 Cel"^^cdt:ucum .

<co> a sosa:Observation ;
  sosa:observedProperty <pipe1#output> ;
  sosa:hasSimpleResult "42.0 Cel"^^cdt:ucum .

<ro> a seas:ComfortEvaluation ;
  seas:evaluationOf <pipe1#output> ;
  seas:evaluatedValue "50.0 Cel"^^cdt:ucum .
```

4.4. Class level properties

Some properties are not suitable for being asserted at instance level. For example, a specific space holds a set of functional and technical requirements that are valid for all instances and a specific type of element such as a project specific brick wall is a container for properties that are valid for all instances of this wall, e.g.: thermal properties, structure etc. Properties like these can be defined as OWL property restrictions. The snippet below shows a project, manufacturer or company specific wall which is defined by property restrictions on its thickness and U-value. The snippet also describes three instances of this wall which all have individual surface areas.

```
ex:HeavyWall rdfs:subClassOf bot:Element ,
  [ a owl:Restriction ;
    owl:onProperty ex:thickness ;
    owl:hasValue "200 mm"^^cdt:ucum ] ,
  [ a owl:Restriction ;
    owl:onProperty ex:uValue ;
    owl:hasValue "0.21 W/K/m2"^^cdt:ucum ] .

<wall1> a ex:HeavyWall ;
    ex:surfaceArea "28 m2"^^cdt:ucum .
<wall2> a ex:HeavyWall ;
    ex:surfaceArea "15 m2"^^cdt:ucum .
<wall3> a ex:HeavyWall ;
    ex:surfaceArea "16 m2"^^cdt:ucum .
```

4.5. Existing BOT implementations

Primary implementations of BOT are reported by Bonduel et al. [10] in datasets, web-applications, or AECOO application plug-ins.

Manual creation of BOT datasets To model existing buildings, one may manually create an ontology that imports BOT. This approach is proposed in [11] and was experimented by different researchers in the W3C LBD CG group while developing BOT. Dedicated user-interfaces could be developed for this, potentially relying on RDF libraries. However, users in the AECOO industry usually use building modelling applications, which implement functionality to export the model as an IFC document.

Export of BOT datasets from IFC documents A converter from IFC documents to BOT, named IFCtoLBD converter, has been developed in the community[18] [10]. This tool extracts instances of bot:Site, bot:Building, bot:Storey, bot:Space, bot:Element and relationships bot:adjacentElement, bot:containsElement, and bot:hasSubElement. Other classes and relationships are not yet supported. In addition to BOT data, IFCtoLBD extracts product, properties, and property values using the OPM ontology [57].

Plug-in for the Revit building modelling application Rasmussen et al. [52] reports on the development of a plug-in for the Revit BIM authoring tool, which leverages the .NET API to export building topology data to a triplestore.[19] The same functionalities as IFCtoLBD are implemented. Moreover, the plug-in has later been developed to export 3D models of spaces and elements as OBJ encoded mesh geometry and outlines of spaces as WKT encoded polygons.

Javascript library for visualising and querying buildings in the browser Rasmussen et al. [52] also reports on the development of a JavaScript library, which can be used to visualise and access building data in the browser.[20] This implementation depended on the Autodesk Forge platform for geometry handling. The Forge viewer uses the Web Graphics Library (WebGL) to render 3D mesh models of zones and elements. In the background, the library issues SPARQL queries to a triple-store to filter the model view, provide table-based results, or colourise zones. Clicking on a zone or an element issues a SPARQL DESCRIBE request with the URI identifying the entity, but could also operate a HTTP GET at this same URI, potentially leveraging the Linked-Data principles. Figure 6 illustrates a mesh geometry generated using the Revit exporter plug-in and visualised in a web browser with a similar JavaScript library.

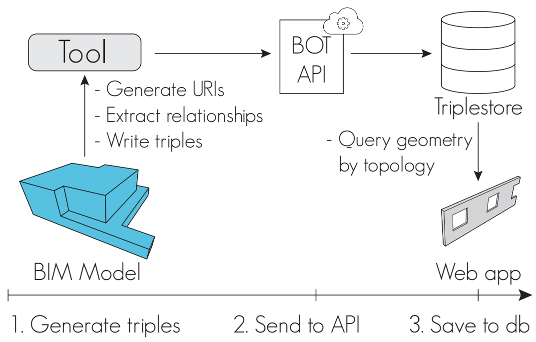

Fig. 10. The infrastructure from triple extraction over the web API to pushing data to the triplestore.

Towards a BIM Maturity Level 3 Linked-Data-based CDE in the browser Figure 10 shows the overall process of getting data from a BIM authoring tool to a triplestore, from where a web application (Figure 6) reads the data. Then, the JavaScript library can combine this data with other sources (i.e. a linked data based CDE). Figure 11 illustrates a demonstration presented in Rasmussen et al. [55], where this library is further extended to integrate building models and sensor observations using SOSA/SSN, allowing to visualise the history of the environmental factors in the browser when clicking on a space, or colouring the spaces according to their current ambient temperature.[21] As these data sources can also be writable, this paves the way for a future decentralised CDE that organically grows a distributed dataset as the design progresses, or during other phases of the life-cycle of the building.

5. Evaluation of BOT and BOT exporters

We already justified throughout Section 3 that the competency questions listed in Section 3.1 are covered by the classes and properties in the BOT

[18] https://github.com/jyrkioraskari/IFCtoLBD
[19] https://github.com/MadsHolten/revit-bot-exporter
[20] demo https://forge-sparql.herokuapp.com/ - sources https://github.com/MadsHolten/forge-sparql
[21] demo - https://youtu.be/P_38gIvrbmg

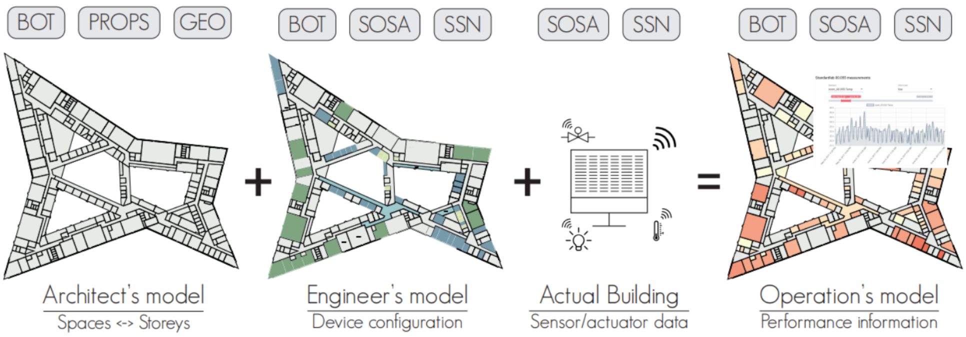

Fig. 11. Visualisation and manipulation of BOT and SOSA/SSN data in the browser. (Illustration from [55])

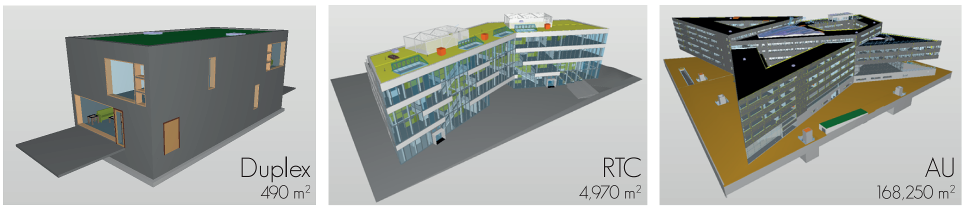

Fig. 12. The three BIM models (Duplex Apartments **[Duplex]**, Technical College in Roskilde **[RTC]**, and the Navitas building at Aarhus University **[AU]**) viewed in Solibri Model Viewer.

ontology. This section provides a supplementary evaluation of BOT on two aspects. Section 5.1 compares the Revit native and IFC exports with the output of the Revit export plug-in introduced in Section 4.5. Then Section 5.2 provides some insight on the BOT reasoning capabilities. Figure 12 illustrates the BIM models on which the evaluations are performed: **[Duplex]** a common BIM file of a 490 m^2 Duplex Apartment;[22] **[RTC]** a 4,970 m^2 Technical College in Roskilde, Denmark; and **[AU]** a 168,250 m^2 university building (Navitas) at Aarhus University, Denmark. The two latter are finalised construction project models by the Danish consulting engineering company Niras.[23] The experiments were performed on a Lenovo P50 laptop with Intel Core i7-6820HQ 2.70 GHz CPU and 32 GB 2133 MHz DDR RAM.

5.1. *Evaluation of the Revit exporter plug-in*

Table 1 summarises the comparison of the exports of (1) the native Revit documents, (2) the IFC STEP Physical File (SPF) documents, and (3) the RDF 1.1 Turtle documents using the Revit exporter plugin introduced in Section 4.5.

The native Revit files are the biggest and are already very well compressed. IFC files are 1.4 **[AU]** to 4.3 **[Duplex]** times smaller, and can further be zipped to an average of 13.5 % of their size. The RDF 1.1 Turtle documents are further 6.6 **[AU]** to 8.5 **[Duplex]** times smaller than the IFC files, and can even be zipped to a smaller average of 8.9 % of their size. Granted, the latter documents contain only a small subset of the information contained in the models, and this subset may grow bigger in future versions of the plug-in. However the exported information is already sufficient to enable the use cases mentioned in Section 4.5.

The export times are evaluated on 5 consecutive exports. Approximately half of the time is dedicated to the generation of geometry (08'36" on average for **[AU]**). In fact, resource-consuming operations such as ray tracing are required to extract high-level topological relationships from the native BIM model.

[22] The RDF export of **[Duplex]** is available on Github (https://github.com/MadsHolten/BOT-Duplex-house), along with a demo application that renders the elements and zones returned by custom SPARQL queries (https://madsholten.github.io/BOT-Duplex-house/)

[23] http://www.niras.com/

Table 1
Comparison of the building model exports for the Duplex Apartments [**Duplex**], Technical College in Roskilde [**RTC**], and the Navitas building at Aarhus University [**AU**]. MB - Megabytes.

	[Duplex]	[RTC]	[AU]
File sizes: Uncompressed (ratio Zipped/uncompressed)			
Revit	10.1 MB (92.4 %)	137 MB (75.9 %)	245 MB (81.4 %)
IFC	2.36 MB (12.9 %)	36.9 MB (13.3 %)	183 MB (11.2 %)
RDF 1.1 Turtle file (plug-in export)	0.278 MB (9.7 %)	6.49 MB (8.0 %)	27.6 MB (8.9 %)
Export with the plug-in as RDF 1.1 Turtle file			
Export time [mm:ss]	00:04.3 ±18 %	00:33 ±6 %	16:14 ±2 %
Number of triples	1,715	20,219	125,973
RDF 1.1 Turtle file: Ratio of the file size (and ratio of the number of triples)			
BOT	17.7 % (53.2 %)	10.7 % (55.0 %)	15.7 % (57.1 %)
Product, properties, property values	13.5 % (29.6 %)	5.4 % (23.0 %)	9.4 % (23.8 %)
Geometry	68.8 % (17.2 %)	83.9 % (22.0 %)	74.9 % (19.1 %)

The plug-in currently does not export all of the BOT axioms that could be exported. For example adjacent elements are only extracted for walls. Some topology relationships are deduced from the native geometry, while they could be deduced from the OBJ objects. The plug-in currently exports a limited set of element product classes (Revit types catalogue, c.f., Section 4.2), and a limited set of properties as simple datatype properties with no units (c.f., Section 4.3). 3D models of zones and elements are exported as mesh geometry OBJ literals, loosing in the process the information regarding the construction process of the geometry.[24] In addition, 2D geometry boundaries of zones is exported as Well Known Text (WKT) literals [14] and linked to the zone with datatype property ex:has2DBoundary. This explains why geometry represents ∼76 % of the file sizes but only ∼20 % of the triples.

5.2. Evaluation of the reasoning on BOT data

In this section we report on the evaluation of six queries that require reasoning capabilities on each of the three building model RDF datasets.

Q1 Select zones (therefore also sites, buildings, storeys, spaces):

```
SELECT * WHERE { ?z a bot:Zone }
```

Q2 Select zones contained in a storey (therefore also the spaces this storey has):

```
SELECT * WHERE
{ ?s a bot:Storey ; bot:containsZone ?z }
```

Q3 Select zones contained in a site (therefore also those transitively contained in the site):

```
SELECT * WHERE
{ ?s a bot:Site ; bot:containsZone ?z }
```

Q4 Select elements contained in a site (therefore also those contained in the zones it contains):

```
SELECT * WHERE
{ ?s a bot:Site ; bot:containsElement ?e }
```

Q5 Select the elements that a site has (therefore also the elements contained in, adjacent to, or intersecting, a zone it contains):

```
SELECT * WHERE
{ ?s a bot:Site ; bot:hasElement ?e }
```

Q6 Select the thickness of each wall.

```
SELECT * WHERE
{ ?e a bot:Element, prod:Wall ;
    props:thickness ?width }
```

[24] Building model software keep track of the operations used to construct the building. For example, (1) define a certain plan, (2) create a point given some coordinates, (3) create a circle in the plan having this point as a centre and a certain radius, (4) extrude the circle along the normal of the plan for a certain length, (5) remove the intersection of the obtained cylinder from another solid, etc.

Table 2

Number of results and query execution times for entailment regime **SL** of Stardog (= **DL** + **SWRL** rules). For the **[AU]** model, execution time for other entailment regimes is provided. Gray indicates best performance between **SL** and **RL** entailment regimes of Stardog for the **[AU]** model. *Note: results for **QL** and **EL** are partial as the queries rely on axioms of BOT that violate this regime.

	Duplex		**RTC**		**AU**				
	#Results	time [ms]	#Results	time [ms]	#Results	execution time [ms]			
		SL		**SL**		**SL**	**DL**	**QL***	**EL***
Q1	27	40	169	170	1,406	940	1,170	970	990
Q2	21	10	146	20	1,392	110	1,090	90	100
Q3	26	10	153	10	1,405	60	40	70	60
Q4	61	20	1,468	10	7,460	350	180	350	360
Q5	102	30	1,858	190	11,183	870	260	920	910
Q6	57	10	976	80	6,181	1,240	140	1,260	1,250

Each query is executed after loading the model in a freshly started Stardog[25] triplestore v5.2.2 to disregard caching optimisation. The process is repeated 10 times to establish mean values.

Table 2 lists the number of results, and the query execution times in milliseconds for entailment regimes (1) **SL** (a combination of DL reasoning and SWRL rules supported by Stardog), (2) **DL**, (2) **QL** (partial) and (3) **EL** (partial). In addition, for the biggest **[AU]** model, execution time for other entailment regimes is provided. Let us note that the transitivity of bot:containsZone violates entailment regime **EL**, so the output results are only partial. As for **QL**, only the axiom **SubClassOf(bot:Interface ObjectMinCardinality(1 bot:interfaceOf))** violates this entailment regime This does not affect the output result for queries **Q1-6** but results are marked with an asterisk. As a conclusion of this evaluation, we argue that the given result times are reasonable enough to rely on BOT and query execution for building user interfaces for web-based CDE, even for large models.

6. Conclusion

The Industry Foundation Classes (IFC) standard is the *de-facto* standard for the file-based exchange of building models between Building Information Modelling (BIM) authoring tools, but there is a need in the Architecture, Engineering, Construction, Owner and Operation (AECOO) industry to evolve to BIM Maturity Level 3, which in essence identifies interoperable and distributed web-based interdisciplinary communication in the AECOO industry. The World Wide Web Consortium (W3C) Linked Building Data (LBD)-Community Group (CG) vision is that the Linked Data (LD) models and best practices can be leveraged for this purpose. In this article, we introduced the Building Topology Ontology (BOT) as the first stable output of this group, and illustrated how BOT is envisioned to be used in combination with other ontologies that describe product catalogues, sensor observation, or IoT devices. We have reported on the current implementations of BOT, and evaluated the export of BOT-compliant Resource Description Framework (RDF) datasets using three native BIM models. The combined use of BOT, existing web-compliant geometry formats, and other ontologies, has been demonstrated in web-based applications. Basic query execution times of less than a second on a building of more than $150,000\ m^2$ demonstrate that using queries over BOT datasets should be suitable for implementing a web-based Common Data Environments (CDEs), thus largely improving the productivity in an AECOO industry where information exchange is currently handled in a predominantly manual, labour-intensive, and error-prone manner.

Although BOT does not alone cover the four general requirements for BIM Maturity Level 3 listed in Section 1, we share the W3C LBD-CG vision that using Linked Data technologies and an open set of well defined ontologies such as BOT is a good direction to be undertaken. In fact:

On REQ1 Using (HTTP) URLs as identifiers for things and making sure that these things are described when looking up those URLs (the

[25] http://www.stardog.com/

three first principles of Linked Data), directly enables information to be exchanged on the Web.

On REQ2 The Web is already used as an information hub for many collaborative, web-based workflows among interdisciplinary stakeholders, not only in the AECOO domain.

On REQ3 The W3C recommendation on Data on the Web Best Practices directly prescribes the use of "terms from shared vocabularies, preferably standardized ones, to encode data and metadata." [41]. Semantic Web technologies are interoperable, flexible, and open, and BOT and other standard and non-standard ontologies can be jointly used to cover different domains.

On REQ4 RDF and the existing ontologies, together with the Linked Data principles, can be used to integrate, for example, building models with openly available datasets (e.g. material property datasets or weather data), and applications (e.g. Geospatial Information Systems (GIS) or Facility Management).

In the future, we will continue to improve BOT, its support in BIM authoring tools and web browser applications, and its integration with other ontologies and datasets. In terms of ontology maintenance the competency questions will be continuously updated. Potential revisions include more detailed topological modelling as introduced by the Region Connection Calculus (RCC) [51]. BOT will be the basis of the development of the W3C LBD CG, which will focus on the interoperable and decentralised web-based description of products and properties, and the homogeneous use of building models across the building life-cycle.

Acknowledgements

After the initial release, BOT is developed continuously as a community effort by the W3C LBD CG. The authors are very thankful for the rewarding discussions in this forum, in the Linked Data in Architecture and Construction (LDAC) workshops and would like to clearly acknowledge all received inputs. We would like to thank Mathias Bonduel (@mathib) and Lucas Verhelst (@LucasVerhelst) as well as the other CG members that have provided human readable labels and comments in their mother tongue to BOT. We would like to thank the reviewers Simon Scheider, Yingjie Hu, and Tomi Kauppinen, for their valuable feedback towards improving this paper, and also Antoine Zimmermann and Iker Esnaola-Gonzalez for their constructive comments on the first version of this article. Part of the work on BOT has been funded by the NIRAS ALECTIA Foundation and Innovation Fund Denmark, the initiative Mittelstand 4.0 by the German Federal Ministry for Economic Affairs and Energy, the French ANR OpenSensingCity project, and ENGIE.

References

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

[2] J. Beetz, J. van Leeuwen and B. de Vries, An ontology web language notation of the industry foundation classes, in: *Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction*, 2005, pp. 193–198.

[3] J. Beetz, J. van Leeuwen and B. de Vries, IfcOWL: A case of transforming EXPRESS schemas into ontologies, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **23**(01) (2008), 89. doi:10.1017/s0890060409000122.

[4] J. Beetz, W. van den Braak, R. Botter, S. Zlatanova and R. de Laat, Interoperable data models for infrastructural artefacts: a novel IFC extension method using RDF vocabularies exemplified with quay wall structures for harbors, in: *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, Vienna, Austria, 2014, pp. 135–140.

[5] T. Berners-Lee, Linked data - design issues, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[6] S. Bertelsen, Construction as a complex system, in: *Proceedings of the 11th Annual Conference of the International Group for Lean Construction*, 2003, pp. 143–168.

[7] M. Bew and M. Richards, Bew-Richards BIM maturity model, in: *BuildingSMART Construct IT Autumn Members Meeting*, Brighton, UK, 2008.

[8] M. Bew, J. Underwood, J. Wix and G. Storer, Going BIM in a commercial world, in: *Proc. of ECPPM*, 2008. doi:10.1201/9780203883327.ch16.

[9] C. Bizer, T. Heath and T. Berners-Lee, Linked Data - The Story So Far, *International Journal on Semantic Web and Information Systems* **5**(3) (2009), 1–22. doi:10.4018/978-1-60960-593-3.ch008.

[10] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen and R. Klein, The IFC to Linked Building Data Converter - Current Status, in: *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*, Vol. 2159, CEUR, London, UK, 2018a, pp. 34–43. ISSN 1613-0073.

[11] M. Bonduel, M.H. Rasmussen, P. Pauwels, M. Vergauwen and R. Klein, A novel workflow to combine BIM and Linked Data for existing buildings, in: *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, Copenhagen, Denmark, 2018b, pp. 407–414. ISBN 978-1-138-58413-6. doi:10.1201/9780429506215-43.

[12] D. Bonino and F. Corno, DogOnt - Ontology Modeling for Intelligent Domotic Environments, in: *Proceedings of the 7th International Semantic Web Conference*, Springer Berlin Heidelberg, Karlsruhe, Germany, 2008, pp. 790–803. doi:10.1007/978-3-540-88564-1_51.

[13] P. Bonsma, I. Bonsma, A.E. Ziri, E. Iadanza, F. Maietti, M. Medici, F. Ferrari, R. Sebastian, S. Bruinenberg and P.M. Lerones, Handling huge and complex 3D geometries with Semantic Web technology, *IOP Conference Series: Materials Science and Engineering* **364** (2018), 1–7. doi:10.1088/1757-899x/364/1/012041.

[14] O.G. Consortium, Well-known text representation of coordinate reference systems (2009).

[15] L. Daniele, F. den Hartog and J. Roes, Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology, in: *7th International Workshop Formal Ontologies Meet Industries*, Springer International Publishing, Berlin, Germany, 2015, pp. 100–112. doi:10.1007/978-3-319-21545-7_9.

[16] J. Davies, *Lightweight Ontologies*, in: *Theory and Applications of Ontology: Computer Applications*, R. Poli, M. Healy and A. Kameas, eds, Springer, Dordrecht, Netherlands, 2010, pp. 197–229. doi:10.1007/978-90-481-8847-5_9.

[17] J. Domingue, D. Fensel and J.A. Hendler (eds), *Handbook of semantic web technologies*, Springer, 2011. doi:10.1007/978-3-540-92913-0.

[18] C. Eastman, P. Teicholz, R. Sacks and K. Liston, *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*, 2nd edn, John Wiley & Sons, Hoboken, USA, 2011, p. 648. ISBN 9780470541371.

[19] J. Euzenat and P. Shvaiko, *Ontology matching*, 2nd edn, Springer, 2014. doi:10.1007/978-1-4614-7163-9_123-1.

[20] Federal Statistical Office of Germany, Distribution of the gross value added in Germany by industries in 2018, 2019a, https://de.statista.com/statistik/daten/studie/252123/umfrage/anteil-der-wirtschaftszweige-an-der-bruttowertschoepfung-in-deutschland/.

[21] Federal Statistical Office of Germany, Number of employees in Construction industry in Germany in 2018, 2019b. https://de.statista.com/statistik/daten/studie/2503/umfrage/anzahl-der-erwerbstaetigen-im-baugewerbe-seit-1992/.

[22] FileFormat.info, Wavefront OBJ File Format Summary, 2017, Last accessed: 02 May 2019. https://www.fileformat.info/format/wavefrontobj/egff.htm.

[23] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari and L. Schneider, Sweetening Ontologies with DOLCE, in: *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, A. Gómez-Pérez and V.R. Benjamins, eds, Springer, Berlin, Heidelberg, 2002, pp. 166–181. ISSN 0302-9743. doi:10.1007/3-540-45810-7_18.

[24] G. Gao, Y.-S. Liu, M. Wang, M. Gu and J.-H. Yong, A query expansion method for retrieving online BIM resources based on Industry Foundation Classes, *Automation in Construction* **56** (2015), 14–25. doi:10.1016/j.autcon.2015.04.006.

[25] G. Gao, Y.-S. Liu, P. Lin, M. Wang, M. Gu and J.-H. Yong, BIMTag: Concept-based automatic semantic annotation of online BIM product resources, *Advanced Engineering Informatics* **31** (2017), 48–61. doi:10.1016/j.aei.2015.10.003.

[26] T.R. Gruber, A translation approach to portable ontology specifications, *Knowledge acquisition* **5**(2) (1993), 199–220. doi:10.1006/knac.1993.1008.

[27] R. Haas, P.J. Keller, J. Hodges and J. Spivak, Quantities, units, dimensions and data types ontologies (QUDT), Technical report, Top Quadrant and NASA, 2014. http://qudt.org.

[28] A. Haller, K. Janowicz, S. Cox, D. Le Phuoc, K. Taylor and M. Lefrançois, Semantic Sensor Network Ontology, W3C Recommendation, World Wide Web Consortium, 2017.

[29] S. Harris, A. Seaborne and E. Prud'hommeaux, SPARQL 1.1 query language, W3C recommendation, World Wide Web Consortium, 2013.

[30] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider and S. Rudolph, OWL 2 web ontology language primer (2009).

[31] ISO 10303-11, Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual, 2004, International Standardisation Organisation, Geneva, Switzerland.

[32] ISO 12006-2, Building construction – Organization of information about construction works – Part 2: Framework for classification, 2015, International Standardisation Organisation, Geneva, Switzerland.

[33] ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries, 2013, International Standardisation Organisation, Geneva, Switzerland.

[34] ISO 19650, Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) – Information management using building information modelling, 2018, International Standardisation Organisation, Geneva, Switzerland.

[35] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik and J. Zhao, PROV-O: The PROV ontology,

W3C Recommendation, World Wide Web Consortium (W3C), 2013.

[36] S.-H. Lee and B.-G. Kim, IFC Extension for Road Structures and Digital Modeling, *Procedia Engineering* **14** (2011), 1037–1042. doi:10.1016/j.proeng.2011.07.130.

[37] M. Lefrançois, Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns, in: *Proceedings of Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT,*, 2017.

[38] M. Lefrançois and A. Zimmermann, The Unified Code for Units of Measure in RDF: cdt:ucum and other UCUM Datatypes, in: *Proc. Extended Semantic Web Conference*, Heraklion, Crete, Greece, 2018. doi:10.1007/978-3-319-98192-5_37.

[39] M. Lefrançois, J. Kalaoja, T. Ghariani and A. Zimmermann, D2.2: The SEAS Knowledge Model, Technical Report, ITEA2 12004 Smart Energy Aware Systems, Brussels, Belgium, 2017.

[40] H. Liu, Y.-S. Liu, P. Pauwels, H. Guo and M. Gu, Enhanced explicit semantic analysis for product model retrieval in construction industry, *IEEE Transactions on Industrial Informatics* **13** (2017), 3361–3369. doi:10.1109/tii.2017.2708727.

[41] B.F. Lóscio, C. Burle and N. Calegari, Data on the web best practices, *W3C Working Draft* **19** (2016).

[42] F. Manola, E. Miller and B. McBride, RDF 1.1 Primer, 2014, Working Group Note, World Wide Web Consortium, 2015.

[43] T. Mendes de Farias, A. Roxin and C. Nicolle, IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties, in: *Proceedings of the 32nd CIB W78 Conference on Information Technology in Construction*, 2015.

[44] M. Niknam and S. Karshenas, A shared ontology approach to semantic representation of BIM data, *Automation in Construction* **80** (2017), 22–36. doi:10.1016/j.autcon.2017.03.013.

[45] P. Pauwels and A. Roxin, SimpleBIM : From full ifcOWL graphs to simplified building graphs, in: *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, Limassol, Cyprus, 2016, pp. 11–18. ISBN 9781138032804.

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

[47] P. Pauwels, S. Zhang and Y.-C. Lee, Semantic web technologies in AEC industry: a literature review, *Automation in Construction* **73** (2017a), 145–165. doi:10.1016/j.autcon.2016.10.003.

[48] P. Pauwels, T. Krijnen, W. Terkaj and J. Beetz, Enhancing the ifcOWL ontology with an alternative representation for geometric data, *Automation in Construction* **80** (2017b), 77–94. doi:10.1016/j.autcon.2017.03.001.

[49] P. Pauwels, M. Poveda-Villalón, Á. Sicilia and J. Euzenat, Semantic technologies and interoperability in the built environment, *Semantic Web* **9**(6) (2018), 731–734. doi:10.3233/sw-180321.

[50] A. Perzylo, N. Somani, M. Rickert and A. Knoll, An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions, in: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Hamburg, Germany, 2015, pp. 4197–4203. doi:10.1109/iros.2015.7353971.

[51] D.A. Randell, Z. Cui and A.G. Cohn, A Spatial Logic Based on Regions and Connection (1992), 165–176–. ISBN 1558602623.

[52] M.H. Rasmussen, C.A. Hviid and J. Karlshøj, Web-based topology queries on a BIM model, in: *5th Linked Data in Architecture and Construction Workshop*, University of Burgundy, Dijon, France, 2017a.

[53] M.H. Rasmussen, P. Pauwels, C.A. Hviid and J. Karlshøj, Proposing a Central AEC Ontology That Allows for Domain Specific Extensions, in: *Proceedings of the Joint Conference on Computing in Construction*, Vol. 1, Heriot-Watt University, Heraklion, Crete, Greece, 2017b. ISBN 978-0-9565951-6-4. doi:10.24928/jc3-2017/0153.

[54] M.H. Rasmussen, P. Pauwels, M. Lefrançois, G.F. Schneider, C. Hviid and J. Karlshøj, Recent changes in the Building Topology Ontology, in: *5th Linked Data in Architecture and Construction Workshop*, Dijon, France, 2017c. doi:10.13140/RG.2.2.32365.28647.

[55] M.H. Rasmussen, C.A. Frausing, C.A. Hviid and J. Karlshøj, Integrating Building Information Modeling and Sensor Observations using Semantic Web, in: *Proceedings of the 9th International Semantic Sensor Networks Workshop, International Semantic Web Conference*, CEUR Workshop Proceedings, Vol. 2213, 2018a, pp. 48–55. ISSN 1613-0073.

[56] M.H. Rasmussen, M. Bonduel, C.A. Hviid and J. Karlshøj, Managing Space Requirements of New Buildings Using Linked Building Data Technologies, in: *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, Copenhagen, Denmark, 2018b, pp. 399–406. doi:10.1201/9780429506215-50.

[57] M.H. Rasmussen, M. Lefrançois, M. Bonduel, C.A. Hviid and J. Karlshøj, OPM: An ontology for describing properties that evolve over time, in: *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*, CEUR Workshop Proceedings, Vol. 2159, CEUR-WS.org, UCL, London, UK, 2018c, pp. 24–33, Accessed September 2018. ISSN 1613-0073.

[58] M.H. Rasmussen, P. Pauwels, M. Lefrançois and G.F. Schneider, Building Topology Ontology, W3C Draft Community Group Report, World Wide Web Consortium, 2020.

[59] C. Reinisch, M. Kofler, F. Iglesias and W. Kastner, ThinkHome Energy Efficiency in Future Smart Homes, *EURASIP Journal on Embedded Systems* **2011**(1) (2011), 104617. doi:10.1155/2011/104617.

[60] G.F. Schneider, Towards Aligning Domain Ontologies with the Building Topology Ontology, in: *5th Linked Data in Architecture and Construction Workshop*, University of Burgundy, Dijon, France, 2017. doi:10.13140/RG.2.2.21802.52169.

[61] G.F. Schneider, Automated Ontology Matching in the Architecture, Engineering and Construction Domain – A Case Study, in: *6th Linked Data in Architecture and Construction Workshop*, CEUR-WS.org, Lissabon, Portugal, 2019, accepted.

[62] G.F. Schneider, A. Bougain, P.S. Noisten and M. Mitterhofer, Information Requirement Definition for BIM: A Life Cycle Perspective, in: *Proc. of ECPPM*, Limassol, Cyprus, 2016.

[63] G.F. Schneider, M.H. Rasmussen, P. Bonsma, J. Oraskari and P. Pauwels, Linked Building Data for Modular Building Information Modelling of a Smart Home, in: *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, Copenhagen, Denmark, 2018, pp. 407–414. doi:10.1201/9780429506215-51.

[64] W. Terkaj and P. Pauwels, A Method to generate a Modular ifcOWL Ontology, in: *Proceedings of the 8th International Workshop on Formal Ontologies meet Industry*, CEUR Workshop Proceedings, Vol. 2050, CEUR-WS.org, Bolzano, Italy, 2017, Accessed September 2018. ISSN 1613-0073.

[65] UNI 11337-5, Building and civil engineering works–Digital management of the informative processes, 2017, Ente Nazionale Italiano di Unificazione, Italy.

[66] P.-Y. Vandenbussche, G.A. Atemezing, M. Poveda-Villalón and B. Vatant, Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web, *Semantic Web* **8**(3) (2017), 437–452. doi:10.3233/SW-160213.

[67] M.P. Villalón and R.G. Castro, Extending the SAREF ontology for building devices and topology (2018).

[68] R. Volk, J. Stengel and F. Schultmann, Building Information Modeling (BIM) for existing buildings: Literature review and future needs, *Automation in construction* **38** (2014), 109–127. doi:10.1016/j.autcon.2013.10.023.

[69] N. Yabuki, E. Lebegue, J. Gual, T. Shitani and L. Zhantao, International Collaboration for Developing the Bridge Product Model "IFC-Bridge", in: *Proceedings of the W78 23rd Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montréal, Canada, 2006, pp. 1927–1936.

[70] C. Zhang, J. Beetz and B. de Vries, BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data, *Semantic Web* **9**(6) (2018), 829–855. doi:10.3233/sw-180297.