Inferential Reasoning in Co-Design Using Semantic Web Standards alongside BHoM

Diellza Elshani[1], Alessio Lombardi[2], Al Fisher[2], Steffen Staab[3,4] , Daniel Hernández[3] and Thomas Wortmann[1]

[1]Chair for Computing in Architecture (CA), Institute for Computational Design and Construction (ICD), Faculty of Architecture and Urban Planning, University of Stuttgart, Keplerstraße 11, Stuttgart, 70174, Germany

[2]Buro Happold, 17 Newman St, London, W1T 1PD, United Kingdom

[3]Department for Analytic Computing (AC), Institute for Parallel and Distributed System (IPVS), University of Stuttgart, Universitätsstraße 38, Stuttgart, 70174, Germany

[4]Electronics and Computer Science, University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom

E-Mail: diellza.elshani@icd.uni-stuttgart.de

Abstract: Design or construction constraints are often considered only in later phases of a linear design process, which leads to costly revisions during construction. Knowledge bases can include logic rules to check constraints and are a powerful tool for representing knowledge on the Semantic Web. Knowledge bases contain facts and rules. The Buildings and Habitats object Model (BHoM) framework, similarly separates objects from methods that assist in deriving knowledge. This paper evaluates data validation, knowledge inferring, and reasoning methods in the Architecture, Engineering, and Construction (AEC) industry. It argues that augmenting BHoM KBs with Semantic Web rules and roles would increase the usage of KBs in the AEC industry by assisting design decisions through inferential reasoning.

Keywords: BIM, Semantic Web, Reasoning, Data Validation, Knowledge Bases

1 Introduction

In current architectural practice, constraints arising from involved disciplines are considered only in later phases of a linear design process. This late consideration often leads to costly revisions shortly before or even during construction. Instead, constraints should be considered as early as possible, which requires representations and reasoning patterns across different kinds of data models[1].

The Semantic Web Ontology Language (OWL) has a rich expressivity and is supported by several reasoning tools [2]. OWL allows for describing the data in terms of concepts and relationships between concept individuals. Furthermore, there are also Semantic Web rule languages which define the operations over the knowledge. While OWL does not define operations because the inference is done by algorithms that have to satisfy the semantics of the ontology, rules indicate explicitly how to perform the inference. These descriptions and rules can be used to check the consistency of the knowledge base and to infer new knowledge. In the Architecture, Engineering and Construction (AEC) industry, the inferred knowledge would recognize design constraints from involved disciplines from early design stages. A Knowledge Base (KB) adds a semantic model to the data by using ontologies and rules for interpreting the data. KBs contain a set of terminological statements (TBox), a set of assertional statements (ABox), and the set of roles defined in the role box (RBox).

The Linked Building Data-Community Group (LBD-CG) [3] is using Semantic Web standards as an open and decentralized alternative to the existing centralized and file-based approaches to storing and sharing data [4]. Many AEC ontologies exist in research and industry, including ifcOWL [5], the OWL representation of the Industry Foundation Classes (IFC) schema [6]. Even though IFC is available in Semantic Web standards and can represent data as a knowledge graph, the support for AEC knowledge graphs and linked data in tool development remains insufficient [7]. In addition, the IFC schema's specification is not based on a logic theory as it was not designed to be translated into rule checking environments [8].

The open-source Buildings and Habitats object Model (BHoM) [9] framework consists of BHoM object models, BHoM Engines, adapters (to map data across design platforms), and user interfaces. Even though BHoM uses an object-oriented data model, its approach to separating objects from functions makes it compatible with ontologies [10]. Whereas BHoM objects (classes and instances) describe facts about the building, which in a knowledge graph corresponds to the facts on TBox and ABox, methods and functions under BHoM Engine can be used to derive knowledge (e.g. properties about object models), which can be found on an RBox in a knowledge graph (Table 1.). Previous research presented the conversion of BHoM object models to knowledge graphs [10]. In this paper, we investigate methods to define axioms constraining roles on BHoM object models based on existing BHoM Engine functions using Semantic Web standards. This investigation requires a comparison of Semantic Web rules that assist to infer new knowledge, roles that define the relationships between objects and properties, and BHoM Engine methods that assist in deriving knowledge from BHoM objects.

Table 1: Similarity of the structure of Knowledge Bases (KB) and BHoM

Knowledge Bases (KB)	**BHoM**
TBox - Terminological Box	BHoM Classes and properties
ABox – Assertional Box	BHoM Instances and property values
RBox – Role box	BHoM Engine

This paper reviews some of the key approaches to data validation and reasoning in the AEC Industry. It discusses the reasoning and data validation methods using Semantic Web technologies as well as derived knowledge from BHoM objects using BHoM Engine methods. Additionally, it argues the rule forms in both approaches, including their syntax and structure. The evaluation and proposal section exemplifies BHoM rules using Semantic Web languages, making BHoM information compatible with OWL reasoners. We apply a semantic reasoner to the resulting graph to discover new facts about the given objects. We conclude by discussing the advantages and limitations of describing BHoM derived properties in Semantic Web languages and present future work possibilities.

2 Review of some approaches to data validation and reasoning in the AEC Industry

In building design processes, designers from different disciplines such as architecture, structure and sustainability collaborate to meet a variety of building performance objectives and constraints [11]. Such objectives and constraints include an appropriate provision of spaces, safety, resource efficiency (e.g., in terms of embodied and emitted carbon dioxide), aesthetics and ease of construction. To consider such performance objectives and constraints as early as possible in design processes, data from different disciplinary data models must be integrated [12]. In the following subsections, we discuss IFC data validation methods, reasoning and data validation using Semantic Web technologies, and the ability to derive knowledge from BHoM objects using BHoM Engine.

2.1 IFC Data Validation

The standard methodology for defining the data exchange requirements and rule constraints for Building Information Models is the Model View Definition (MVD) [13]. MVD specifies the subsets of an Industry Foundation Classes (IFC) schema, including entity, attribute, and geometry representation constraints. BuildingSMART recommends the use of mvdXML as a formal representation format for MVDs. Model Checking in BIM includes (1) BIM validation which checks modelling attributes and procedures, (2) clash detection, i.e. interference check; and (3) code checking, verifying compliance with the correspondent regulation. While MVD checking focuses on fast validation of data structures and values in raw IFC data, semantic rule checking methods for BIMs focuses on enriching geometry calculation and semantic inferencing [8]. Model Checking in BIM using the IFC standard might be very powerful in documenting and storing design data; however, IFCs' hierarchical, heavy and monolithic data model makes it complex to be used during the design phase. However, the most critical decisions in building design are made in the conceptual design phase, and they influence not only construction costs but also building energy consumption [1]. Lack of data validation and reasoning during the whole design phase, including the early stages of design, may prevent the recognition of violated design constraints until it is too late, i.e., until construction has already started. Additionally, when considering a logic-based rule checking environment for the AEC industry, one must consider its source of information first [8]. The IFC schema's specification is not based on a logic theory because it was not specifically designed for import into rule-checking environments [8].

2.2 Reasoning and Data Validation Using Semantic Web Technologies

Instead of relying on document-based building models, a promising approach for enhancing interoperability with integrated data is the use of the Semantic Web, [14] ,[3]. Semantic Web technologies allow for cross-domain linking, advanced regulations and rule set checking as well as reasoning on data.

The Semantic Web covers a set of technologies; the standard data model of the Semantic Web technology stack is the Resource Description Framework (RDF) [15]. In 2017, the W3C proposed Shapes Constraint Language (SHACL) [16], as the language to validate data in the RDF model [Check]. SHACL defines the shape of the RDF data. SHACL define constraints with expressions called shapes. Each shape consists of a name, a restriction, and an expression determining a set of resources in the data that have to satisfy the shape. OWL is a formal language for authoring ontologies [2]. OWL allows for expressing complex concept definitions, relations between concepts and roles, and inferring new knowledge from these definitions. SHACL covers data validation and constraints and is one of the Semantic Web technologies that assume a closed world [16].With a Closed World Assumption (CWA), any statement that is not known to be true is considered false. Many conventional design software applications adopt a CWA, including BIM tools and common database systems [17]. OWL assumes an open world, where missing information is simply unknown. For instance, with the aforementioned axioms (2) and (3) and the assertion stating that rp is a RoofPanel (i.e., ClassAssertion(RoofPanel rp)), if rp has no explicit ID, a reasoner my produce a new element rpID to represent the identifier of rp (i.e., the assertions ObjectPropertyAssertion(hasID rp rpID) and ClassAssertion(Identifier rpID) are inferred). While SHACL constrains data to follow a schema, OWL provides additional inferencing. In other words, reasoning allows inferring new knowledge, and data validation with SHACL allows seeing inconsistencies by indicating that there are design violations. Ontology reasoners can assist in ontology consistency, class satisfiability, classification, instance checking, and conjunctive query answering.

The Semantic Rule Language (SWRL) is a W3C standard that combines OWL and the Rule Markup Language (RuleML). In SWRL rules are expressed in terms of OWL concepts, including classes, properties, and individuals. Rules define new assertions as a consequence of previous assertions. For example, the rule:

hasColumns(?x,?y) ∧ columnsMaterialIsTimber (?y,?z)⇒ isTimberBuilding(?x,?z)

says that if an object ?x has columns ?y and these columns ?y are of timber material ?z, then ?x is a timber building of material ?z.

2.3 Deriving knowledge from BHoM objects using BHoM Engine methods

Although BHoM employs an object-oriented data model, its separation of object models from data-processing functions brings it closer to ontologies and knowledge bases [10]. All functionality applicable to the *object models (oM)* types is isolated, and it is primarily grouped in C# Projects called *Engines*. Similarly to *oM* projects, engine projects target a specific domain, use specific namespaces and are suffixed with "*Engine*", e.g. methods for Structural Engineering are placed in the *Structure_Engine* under namespaces starting with *BH.Engine.Structure*. Engines are essentially collections of five different kinds of static classes (Table 2.), which are used as groups for the methods. Each function must clearly target one main input type, and methods are to be written as extension methods, so they are available throughout the framework as an extension to the type. This makes the oM C# classes work in a manner that is closer to dynamically typed languages (e.g. Python) than statically typed languages, and is a design choice that simplifies contribution and scalability. Each oM type can effectively be augmented with precise functions included in Engines. BHoM objects describe facts about the building, which correspond to TBox and ABox axioms in a knowledge graph. BHoM Engine methods and functions can be used to derive knowledge, such as properties about object models, from an RBox in a knowledge graph. Query methods in *BHoM_Engine* are qualified as derived properties and also check certain rules. For example, the centreline of a Bar is an example of a derived property of a Bar. The Bar does not have a Centreline as a declared property (directly defined in the Bar class), but it has *StartNode* and *EndNode*. The centreline can therefore be derived from a function that returns the centreline of the Bar as the line between the *StratNode* and *EndNode* (See Fig. 1). This function is a function that lives under the BHoM Engine. Similarly to other properties, for example, the extrusion is a derived property. Given the declared properties, it is possible to first compute the *Centreline* of the bar, then using its cross-section property one can extrude the cross-section along the centreline, getting the extrusion.

Table 2: BHoM Engine Classes

Create class	To instantiate types (similarly to a singleton pattern)
Modify methods	Operate on the data stored on instances
Query	To retrieve derived data from the types
Convert	Changes a type for another
Compute	Hosts computationally intensive function

```
public virtual Node StartNode { get; set; }
        public virtual Node EndNode { get; set; }
        public virtual ISectionProperty SectionProperty { get; set; } = null;
        public virtual double OrientationAngle { get; set; } = 0;
        public virtual BarRelease Release { get; set; } = null;
        public virtual BarFEAType FEAType { get; set; } =
        BarFEAType.Flexural;
        public virtual Constraint4DOF Support { get; set; } = null;
         public virtual Offset Offset { get; set; } = null;
```

```
public static Line Centreline(this Bar bar)
        {
        Ret rn bar.IsNull() ? null : new Line { Start = bar.StartNode.Position,
        End = bar.EndNode.Position
                };
```

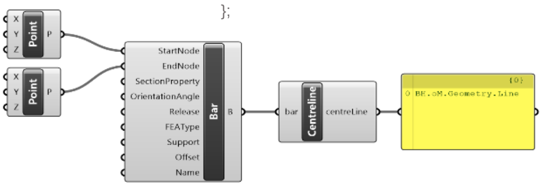

Figure 1: (Left) A bar, structural namespace of the BHoM object models; (right) the function to derive its centerline as a derived

3 Evaluation and Proposal

In this section, we evaluate the usage of Semantic Web standards to validate data and infer knowledge in AEC and BHoM Engine that similarly assist in deriving new knowledge from objects.

3.1 Evaluation of reasoning and data validation in AEC

Using Semantic Web technologies, one can validate data through schemas or apply to reason about a given knowledge base. While the term linked data is closer to "web of data", the term "Semantic Web" encompasses all aspects of the Semantic Web stack, SHACL, OWL, rules, proofs, and truth [17]. The linked data principles of Berners-Lee [18] provide a solid ground for data interoperability for heterogeneous data sources and allow for querying complex questions. However, the principles do not include ontologies, rules and proofs. Consequently, the linked data emerging out of this proposal leaves aside some Semantic Web technologies [17]. This neglect might have caused a low usage of reasoning and methods in applications of linked data. Current building data, rules and restrictions for data validation (e.g. MVDs) do not rely on the same environments, so interoperability problems between tools often hinder reasoning and data validation. Such data validators are usually used only at the end of design processes. Using Semantic Web standards to link data, designers could benefit from the power of knowledge graphs, including discovering new knowledge, reasoning, and data validation. In short, the potential of Semantic Web is not fully unlocked in the AEC industry. Since BHoM provides user interfaces in many design software, using BHoM to model data and define specific rules that can infer new knowledge and convert it to Semantic Web standards would make knowledge graphs more accessible to architectural designers. With such infrastructure, designers could validate design options in real-time, and design with ontologies and rule-checking constraints in their design platforms (e.g. Grasshopper 3D). BHoM allows integration of both OWA as well as CWA. While OWA should reflect the incremental nature of design processes, (partial) CWA reasoning should allow for discovering lacking specifications. Beetz emphasized that as long as the geometric information is not used in a logical inference process, an RDF representation of the building information is inefficient and does not add much additional value [20]. BHoM allows geometric representation of building elements using primitive data types (e.g. it represents NURBS curves using the coordinates of control points). Introducing algebraic formulas for geometries and adding rules that define the relations between these geometries will make a step forward towards using geometric information in logical inferences. Therefore, integrating geometric information in logical inferences is possible with BHoM.

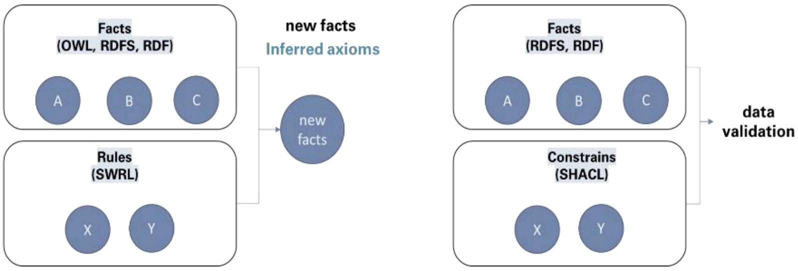

Figure 2. Left: Inferring new knowledge with reasoning; Right: validating data with SHACL shapes.

3.2 Example: Using Semantic Web rules with BHoM Engine methods

BHoM information can be converted to SWRL as well as SHACL, based on the purpose of the design. For example, calculating a centerline length of a bar from its Start Node and End node using SWRL language could be described as follows:

Column (c1) ∧ StartNode (c?, ?p1) ∧ EndNode (?c, ?p2) ∧ subtract (?Length, ?p2, ?p1) ⇒ Length (?c, ?length).

With a reasoner one can infer the length of the centerline, based on the given StartNode and EndNode variables. In the given example, for simplicity, we do not show the datatypes of the SartNode and EndNode, and consider that they are xsd:dobule. On the other hand, if we want to make sure that every bar has at least one start node, we could use SHACL:

BhomBarShape a sh:NodeShape; sh:targetClass :Bar ; sh:property [sh:path :hasStartNode ; sh:minCount 1 ; sh:node :StartNodeShape.

Both approaches could support the BHoM knowledge graph and allow one to infer new knowledge with reasoning mechanisms or validate data against SHACL definitions to satisfy a given set of requirements.

4 Conclusion and future work

This paper discusses and evaluates data validation and reasoning methods in the AEC industry. It discusses how to integrate explicit knowledge and implicit building information (rules) into a KB. It also presents the advantages and potentials of BHoM ontologies, operators and reasoning methods, which could offer a novel knowledge representation framework for co-design processes in the AEC industry. We conclude that there are several approaches to append new information on a BHoM KB using either BHoM Engine, RBox roles, or Semantic Web rules. Future work will analyse when one or the other of these approaches should be used by providing advantages and disadvantages of each approach based on the tasks that need to be completed. Combining the BHoM framework with Semantic Web standards can increase the use of knowledge graphs in the AEC industry, by not only improving data interoperability but also assisting design decisions through inferential reasoning.

Acknowledgements

Partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2120/1 - 390831618 and COFFEE - STA 572_15-2.

References

[1] D. Elshani, T. Wortmann, and S. Staab, “Towards Better Co-Design with Disciplinary Ontologies: Review and Evaluation of Data Interoperability in the AEC Industry.,” Linked Data in Architecture and Construction 2022 ESWC, 2022.

[2] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, “OWL 2 Web Ontology Language Primer” W3C Recommendation, 2021. https://www.w3.org/TR/owl2-primer/

[3] P. Pauwels, A. Costin, and M. H. Rasmussen, “Knowledge Graphs and Linked Data for the Built Environment,” in Industry 4.0 for the Built Environment, vol. 20, M. Bolpagni, R. Gavina, and D. Ribeiro, Eds. Cham: Springer International Publishing, 2022, pp. 157–183

[4] P. Poinet, D. Stefanescu, and E. Papadonikolaki, “Collaborative Workflows and Version Control Through Open-Source and Distributed Common Data Environment,” in Proceedings of the 18th International Conference on Computing in Civil and Building Engineering, vol. 98, E. Toledo Santos and S. Scheer, Eds. Cham: Springer International Publishing, 2021, pp. 228

[5] P. Pauwels and W. Terkaj, “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology,” Automation in Construction, vol. 63, pp. 100–133

[6] buildingSMART.“Industry Foundation Classes.bSI Standards”n.d. buildingsmart.com

[7] P. Pauwels, S. Zhang, and Y.-C. Lee, “Semantic web technologies in AEC industry: A literature overview,” Automation in Construction, vol. 73, pp. 145–165, 2017

[8] P. Pauwels et al., “A semantic rule checking environment for building performance checking,” Automation in Construction, vol. 20, no. 5, pp. 506–518, Aug. 2011

[9] “BHoM. The Buildings and Habitats object Model,” https://bhom.xyz/, n.d.

[10] D. Elshani, A. Lombardi, A. Fisher, D. Hernandez, S. Staab, and T. Wortmann, “Knowledge Graphs for Multidisciplinary Co-Design: Introducing RDF to BHoM,” Linked Data in Architecture and Construction 2022, ESWC 2022.

[11] T. Wortmann and G. Nannicini, “Introduction to Architectural Design Optimization,” in City Networks, vol. 128, A. Karakitsiou, A. Migdalas, S. Th. Rassia, and P. M. Pardalos, Eds. Cham: Springer International Publishing, 2017, pp. 259–278.

[12] T. Wortmann and B. Tunçer, “Differentiating parametric design: Digital workflows in contemporary architecture and construction,” Design Studies, vol. 52, pp. 173–197, Sep. 2017

[13] H. Liu, G. Gao, H. Zhang, Y.-S. Liu, Y. Song, and M. Gu, “MVDLite: a Fast Validation Algorithm for Model View Definition Rules.” arXiv, 2019.

[14] A. Hogan et al., “Knowledge Graphs,” Synthesis Lectures on Data, Semantics, and Knowledge, vol. 12, no. 2, pp. 1–257, Nov. 2021

[15] D. Brickley and R. V. Guha, “RDF Schema 1.1.” 2014. [Online]. Available: https://www.w3.org/TR/2014/REC-rdf-schema-20140225/

[16] H. Knublauch, “Shapes Constraint Language (SHACL),” 2017. https://www.w3.org/TR/shacl/

[17] P. Pauwels, S. Zhang, and Y.-C. Lee, “Semantic web technologies in AEC industry: A literature overview,” Automation in Construction, vol. 73, pp. 145–165, Jan. 2017

[18] T. Berners-Lee, “Note on Linked Data,” 2006. https://www.w3.org/DesignIssues/LinkedData.html