A domain ontology for representing core modular building components aligned with Linked Building Data and Design for X principles

Salik Ahmed Khan[1], Yang Zou[2], Minh Kieu[1]

[1]Department of Civil and Environmental Engineering, University of Auckland, New Zealand
[2]School of Architecture, Building and Civil Engineering, Loughborough University, United Kingdom
skha864@aucklanduni.ac.nz, y.zou2@lboro.ac.uk, minh.kieu@auckland.ac.nz

Abstract –

Design for manufacturing and assembly (DfMA), extended through more generic Design for X (DfX) is an important component for realising the benefits of modular and industrialised construction. This requires balancing heterogeneous objectives derived from diverse, often fragmented data sets. In the absence of a unified data model, designers often resort to bespoke solutions on a per-case basis, missing the opportunity for optimally mass-customised solutions. Semantic web technologies have proven the capability to unify diverse data sources through the formalisation and re-use of knowledge across lifecycles and projects. But its application within modular construction has remained limited. This paper proposes the Modular Building Components Ontology (MBCO) to formally represent components' hierarchy in line with Linked Building Data (LBD) and DfX principles. The Ontology 101 methodological framework is adopted. It reuses and aligns high-level vocabularies like Building Topology Ontology (BOT) to enhance interoperability. It provides a modular core structure with optional extensions to further knowledge (like environmental indicators, constructability, mass customisation, etc.) without bloating the schema. Reasoning-based checks and SHACL shapes ensure consistency and data quality. We illustrate feasibility through a case study that populates a residential modular unit and show how SPARQL queries and lightweight rules can extract objective metrics like component commonality and production-assembly constraints directly from the graph. The results show that such an ontology can facilitate automated simultaneous multi-objective optimisation using diverse data sources, thereby operationalising DfX in modular construction.

Keywords –

Modular Buildings; Ontology; Linked Building Data; Design for X

1 Introduction

Modular and industrialised building construction has promised a huge potential with the recent advancements in the architecture, engineering, and construction (AEC) industry. The benefits realisation, however, has remained limited due to several challenges, design being one of the foremost. This is because the industrialised construction process largely relies upon modular design properties to achieve benefits through mass customisation. The disjointed process of offsite production and onsite assembly makes the consideration of downstream requirements within design more important, giving rise to the Design for Manufacturing and Assembly (DfMA) philosophy. More recently, this has been replaced by a notion of Design for X (DfX) [1], where the considerations are expanded to sustainability, modular suitability, etc. Multi-objective design optimisation, akin to the manufacturing domain, has thus gained momentum to address this gap [2]. The problem with that, however, is the fragmented nature of information linked to multiple objectives. For example, components' environmental data, manufacturing process, module transportation requirements, and crane assembly constraints all have different standards for representing information, and real-time inclusion to support the designer is often difficult, if not absent at all [3,4]. Semantic web technologies provide a mechanism to unify such information into a consistent model through the Linked Building Data (LBD) initiative for the AEC domain [5]. Existing applications within modular construction have primarily focussed on standardising the DfX-based design review process. However, the automation of optimal design generation promises wider benefits in view of the recent AI-based methods. These methods heavily rely on the representation of design, with graph-based methods showing promising results [6]. However, no formalisation of a graph-based modular building component structure could be found which would streamline these efforts and exploit the benefits of standardisation. A need, thus, arises to formalise a core representation of modular buildings within the LBD context, based on which the requirements for automated design generation may be uniformly defined.

This paper tries to address this gap through the development of a core ontology to represent modular building component structure, aligned with existing efforts and allowing for subsequent extensions to support a unified design optimisation workflow. Ontology development 101 methodological framework provides a comprehensive, step-by-step guide for this process, and has been utilised for the current work. The Web Ontology Language (OWL) is used for the specifications, and it works on the Resource Description Framework (RDF) data model. The proposed ontology is meant to be positioned as middle-to-upper level, formalising domain-specific concepts, while aligning with existing high-level definitions like Building Topology Ontology (BOT) [7] for building context and spatial containment, and Ontology for Property Management (OPM) [8] for component properties. Core classes like Volumetric Module, Planar Module, etc. extend BOT concepts, while allowing for further domain extensions to suit multiple DfX criteria. Similarly, a set of minimal attributes are associated with the classes, which may be easily extended to capture details like geometry, materials, manufacturer details, etc. A modular architecture is employed to define a core structure, which is extendable through Linked Data hooks to external knowledge bases, without bloating the schema. Manufacturability and Mass Customisation based DfX criteria definitions are further constructed to demonstrate usability and automated reasoning across domains. Shapes Constraints Language (SHACL) has been used to define rules for consistency and manufacturability to ensure data quality and real-world applicability. The feasibility is illustrated through populating a residential modular building in line with the ontology and computing objective values of multiple criteria like component commonality, module dimensions, and weight limits from the RDF graph.

2 Background

The current work focusses on two primary theoretical foundations to develop the case for this research – (1) Design within modular buildings and the various considerations required for achieving the promised benefits of modular construction; (2) The application of semantic web and linked data technologies within modular construction to implement the proposed design principles.

2.1 Design in Modular Construction

Based on the requirements for design in modular construction, two distinct considerations can be pointed out – DfMA and Mass Customization.

DfMA involves incorporating downstream manufacturing and assembly requirements during design and originated in manufacturing. In AEC, its integration with BIM has been shown to improve process flow, although regulatory, cultural, and knowledge-related barriers remain [9]. These have also identified DfMA's potential for autonomous design and construction manufacturing, while demonstrating that existing work places greater emphasis on design review than on the a priori inclusion of design requirements, consistent with Lee et al. [1]. A broader Design for X (DfX) approach has further been proposed, where "X" may represent criteria such as modularity, sustainability, commonality, variety, production cost, transportability, etc. In this form, DfX aligns more closely with mass customisation as compared to DfMA.

Mass customisation seeks to combine the efficiencies of mass production with a desired level of variety. It is commonly enabled through product platform architectures and the customer-oriented design decoupling point model. Jensen [10] identified four modular construction platform strategies based on the decoupling point: engineered-to-order, modify-to-order, configured-to-order, and select-variant-to-order, in increasing order of standardisation. This trade-off between standardisation and customisation is central to DfX. The present work addresses its formalisation through design representation.

2.2 Linked Data for Modular Buildings

Semantic web and linked data technologies have shown promise in addressing information fragmentation in AEC, leading to standard knowledge dictionaries for representing spatial topology, product structure, geometry, and element properties [7,8,11]. For modular buildings, however, although product-based structuring has been recognised at a high level, domain-specific formalisation remains limited. Early efforts include the Product Semantic Representation Language (PSRL) and semantic web-based component catalogue representations highlighting the need for standardised components [12]. Subsequent studies proposed semantic support for DfMA-based design, manufacturing integration, and cost estimation [13]. Lee et al. [1] similarly aimed to standardise DfX review for modular buildings, but focused on process concepts without adequately representing the design artefact itself.

Other work has linked DfMA with mass customisation through client-engagement modelling [14], or proposed ontologies for constructability and mass customisation [15,16]. However, these approaches are either too component-centric for inter-domain DfX use or limited to lower-level product library descriptions

Table 1: Comparison of MBCO with existing efforts for formalised graph-based modular building representation. MLE=Modular, Lightweight, and Extensible Ontology Structure

References	Ontology Proposed	LBD Alignment	MLE Structure	DfX Criteria Inclusion	Abstraction Level
Gan [7]	No	No	No	Multi	-
IfcOwl [18]	Yes	Yes	No	No	All
DfX [1,14,15,20]	Yes	No	No	Single	Middle
Cao et al. [20]	Yes	Yes	No	Single	Lower
Rasmussen et al. [18]	Yes	Yes	Yes	No	Lower
MBCO	Yes	Yes	Yes	Multi	Middle

despite alignment with ifcOWL [17,18]. Further ontology-based manufacturability studies [19,20] define core component concepts, but do not adequately capture module–element compositional relationships or align well with existing linked building data ontologies. A lean, comprehensive, and standardised cross-domain design representation is therefore still lacking.

2.3 Research Gaps

The following research gaps can be summarised based on existing literature:

1. There is no standard ontology for representing modular buildings' component hierarchy concepts. Particularly, an alignment with upper-level ontological frameworks like BOT, BPO, etc. is missing.
2. There is a lack of a middle-level domain specific ontology definition for modular buildings. The existing efforts on modular building ontologies target the lower-level definitions, often based on individual cases or specific component instances, aligning more with the definition of a knowledge graph rather than an ontology. This also limits the multi-domain reasoning capability for DfX as each study defines all definitions based on its own singular criterion (e.g., manufacturability [19], constructability [15], cost assessment [13], etc.).
3. Most work on ontology-based DfX for modular buildings is focussed on the design review process [1], and not on design generation process, which limits the automation of design.

To address these gaps, this study develops the Modular Building Component Ontology (MBCO) as a federated extension to BOT in the domain of modular construction, aligning with the principles of domain extensible shared ontology [21], and Linked Data deployment scheme. This is meant to compliment the work of LBD Community Group[1] of the World Wide Web Consortium (W3C) in the direction of modular buildings. Table 1 compares MBCO with existing efforts aimed at formalising a graph-based representation for modular buildings across five dimensions. Here, LBD alignment assesses the reuse of existing ontologies. The modular, lightweight, and extensible (MLE) criterion assesses the ontology structure against linked open data principle of defining ontologies as light-weight modules which are easy to maintain and extend as required.

The positioning of the current study relative to the wider ontology development efforts within the AEC domain is illustrated in Figure 1. In this context, MBCO is meant to be a bridge connecting upper-level AEC domain ontologies with lower-level ontologies built for specific objectives.

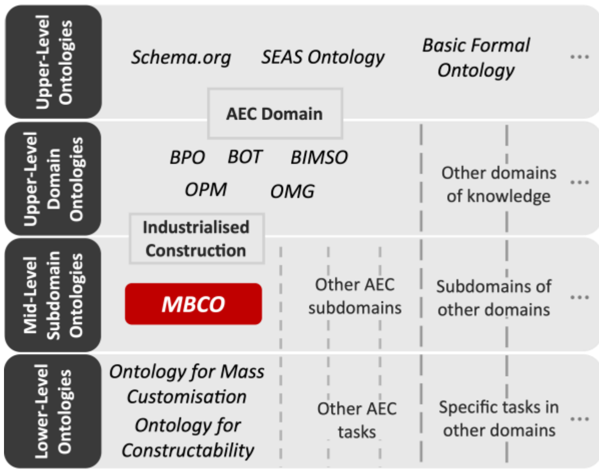

Figure 1. Positioning of MBCO among the wider ontology development efforts.

3 Methodology

The development of the proposed ontology follows a six-step process. This is guided by the Ontology Development 101 framework [22]. This method was selected because of its widespread use within

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

construction applications and the provision of a tool-supported, step-by-step procedure incorporating best practices for ontology building [4,22]. This includes: (1) the definition of domain, scope, and competency questions, (2) reuse and alignment with existing ontologies, (3) enumerating key terms, (4) defining classes, (5) class properties & their types, and (6) creating instances and evaluation of ontology. Protégé is an open-source tool recommended by Ontology Development 101, and is used here, with Web Ontology Language (OWL) based reasoning to verify logical consistency and inferred classifications. The resulting ontology is organised in a modular manner – a lean core schema for component hierarchy and connectivity, with optional extensions to external datasets (like environmental indicators, etc.) to avoid over-complicating the core vocabulary while enabling multi-domain reasoning workflows.

4 Modular Building Component Ontology

4.1 Domain, Scope & Competency Questions

MBCO is developed to provide a core formal, reusable representation of modular building component hierarchy that supports design reasoning and automation, and which can be extended with specialised DfX ontologies. Here we use the term component in a loose definition to refer to any part of a modular building.

The scope of the ontology is restricted to defining (i) a core modular component decomposition structure, and (ii) a minimal set of extendable attributes for multi-criteria DfX incorporation. To further clarify the scope, the following competency questions (CQs) are developed:

CQ1. What are the types and numbers of all modules that constitute a given modular building?
CQ2. What smaller modules and/or elements are contained inside a larger module?
CQ3. What modules are adjacent to or intersecting with a given module?
CQ4. How to assign metadata to modules, elements, or their connections?
CQ5. How to assign attributes to constituents of a modular building to serve a given DfX criterion?
CQ6. How can attributes of modules/elements be reused across multiple DfX criteria?

Further, separate modular ontologies can be integrated for each DfX criterion that needs to be considered. Here, we show this by the examples of manufacturability and mass customization used in combination with the core MBCO. For example, competency questions for assessing mass customisation criteria using MBCO may be considered as follows:

CQ7. Which components are reused across multiple modules and with which properties?
CQ8. What is the aggregated manufacturing cost based on a given modular composition of a building?

4.2 Reuse of Existing Ontologies

Reuse of and alignment with existing ontologies and knowledge definitions is one of the key principles of linked data implementation [22]. To avoid duplicate effort, existing libraries of reusable ontologies like Linked Open Vocabularies[2], DAML[3], and Ontobee[4] were searched for terms related to modular buildings, industrialised construction, prefabrication, and design. Further, Scopus[5] database was searched for relevant works of a non-ontological nature, or which might not be published as open use ontologies. Although some recent efforts have tried to implement ontology-based design representations for DfX [15,19], but no effort was found providing a core, reusable modular building structure, in-line with W3C linked data deployment scheme which is reusable across multiple DfX criteria.

To ensure interoperability with the LBD ecosystem, MBCO reuses and aligns established vocabularies rather than redefining generic building concepts related to topology, products, or geometry. BOT is adopted to represent building topology, including zones, elements, and their containment and adjacency relationships. This caters to partially address CQ2, CQ3, and CQ4. Further, BPO is aligned to represent part-whole assembly relationships addressing CQ2. For the core concepts of MBCO, the product decomposition taxonomies from [6,23] and ontology concepts from [15,19,20] are reused to varying extents to address CQ1 through CQ6, as explained below. To attach metadata, we utilise concepts from ontology for property management (OPM) and ontology for manging geometry (OMG) to address CQ4.

4.3 Enumerating Important Terms

Important terms related to modular buildings' component hierarchy have been identified through knowledge sources including textbooks , articles, and open-source data dictionaries describing modular design [23–25]. Continuing from the scope definition, the important terms for MBCO may be viewed from two interlinked perspectives – (i) terms representing the core modular product decomposition structure in view of CQ1 through CQ4, like modular building, volumetric module, planar module, framing element, connection, interface, material, geometry, typology, sub-components, etc., and (ii) terms allowing for multiple DfX criteria inclusion in

[2] https://lov.linkeddata.es/dataset/lov
[3] https://www.daml.org/ontologies/
[4] https://ontobee.org/
[5] https://www.scopus.com/

view of CQ5 and CQ6, like DfX criterion, DfX performance indicator, etc.

4.4 Defining Classes and Class Hierarchy

The classes and their hierarchy are defined based on the enumerated concepts and the reused ontologies. A combination development process [22] is employed, which starts with the description of the most salient concepts like *VolumetricModule* and *PlanarModule*. These are then generalised (bottom-up) into a *Module* class and are specialised (top-down) into *ModuleType* class. *ModuleType* is introduced here as a generic concept to distinguish modules at any level of hierarchy. For example, *PlanarModule* may be specialised into Exterior wall, Interior wall, Slab, Framing, etc., or *VolumetricModule* may be specialised into Bathroom pod, Kitchen pod, or Balcony module, etc. But these concepts are left to be decided by the user, in this case, the designer or manufacturer. This is in line with addressing CQ1 and CQ6 and maintaining MBCO as a middle level domain ontology. *VolumetricModule* is then aligned as a subclass of *bot:Zone*, while *PlanarModule* is specified under *bot:Element*. The BOT classes allow the topological expressivity to address CQ2 and CQ3. Additionally, *mbco:ModularBuilding*, a subclass of *bot:Building* is deemed necessary to allow for building level attributes specific to modular buildings. Base attributes of a *Module* like material, geometry, etc. are supported directly through *opm:Property* class, which may further be linked to *omg:Geometry* class [11], addressing CQ4. Similarly, the types of connections between modules, elements, and modules and elements play an important role in defining the component breakdown structure in modular buildings, often influencing design choices related to mass customisation, cost, and constructability. *Connection* class has been defined here to specify these requirements as a subclass of *bot:Interface*. Further, *ConnectionType* class is defined in relation to *Connection* class to fully address the needs for CQ4.

Similar process is repeated for defining the DfX related classes addressing CQ5 and CQ6, where *DfXCriterion* represents a given module's DfX criterion under assessment, like manufacturability, constructability, etc. *DfXIndicator* class further encapsulates the performance indicators for individual DfX criteria like dimensional constraints for manufacturability, lifting constraints for constructability, typology constraints for mass customisation, etc. of a given module or element. These classes need to be utilised in combination with the task-specific ontologies extending the MBCO for realising comprehensive assessments.

4.5 Defining Properties and their Types

Properties describe the internal structure of the classes and are essential to realising the competency questions [22]. Object properties link classes to other classes, while datatype properties link classes to specific values as necessary. To operationalise the class model in MBCO, we define a small set of object properties deemed essential to answer the competency questions, as summarised in Table 2:

1. Modular decomposition across building levels is anchored in BOT. *mbco:hasModule* links a *bot:Storey* to any *mbco:Module*, while *mbco:hasVolumetricModule* specialises this for *mbco:VolumetricModule* and is declared a subproperty of both *mbco:hasModule* and *bot:containsZone*; similarly, *mbco:containsPlanarModule* is a subproperty of *mbco:hasModule* and *bot:hasElement*, enabling query-compatible retrieval while preserving BOT semantics.
2. *mbco:hasProperty* connects a *Module* to *opm:Property* definition for allowing attributes like

Table 2: Summary of object properties in MBCO (all entities defined in *mbco:* domain except specified otherwise)

Competency Addressed	Object Properties	Sub-Property of	Domain	Range
CQ3, CQ4	*connectionOf*	*bot:interfaceOf*	*Connection*	*Module, bot:Element*
CQ1, CQ2	*containsPlanarModule*	*hasModule, bot:hasElement*	*bot:Zone*	*PlanarModule*
CQ4	*hasConnectionType*	-	*Connection*	*ConnectionType*
CQ5, CQ6	*hasDfXCriterion*	-	*ModularBuilding, Module*	*DfXCriterion*
CQ5, CQ6	*hasDfXIndicator*	-	*DfXCriterion*	*DfXIndicator*
CQ1	*hasModule*	-	*bot:Storey*	*Module*
CQ1	*hasModuleType*	-	*Module*	*ModuleType*
CQ4	*hasProperty*	*seas:hasProperty*	*Module*	*opm:Property*
CQ1, CQ2	*hasVolumetricModule*	*hasModule, bot:contaisZone*	*bot:Storey*	*VolumetricModule*

material, geometry, etc. Where the extended hierarchy of properties is not required, *props* predicates can be directly used from IFC models as done by the IFC-to-LBD convertor [26].

3. Inter-module connectivity is captured via *mbco:connectionOf* to represent realised interfaces between modules.
4. Explicit type assignment relations are provided using *mbco:hasModuleType* and *mbco:hasConnectionType*. These properties support integration with controlled vocabularies (e.g., manufacturer catalogues, platform families, standard connection taxonomies). In practice, *ModuleType* and *ConnectionType* can evolve into enumerated classes or be aligned to higher level concept schemes (like *skos:Concept*), depending on whether strict logical typing or flexible terminological tagging is desired.
5. DfX assessment is modelled using *mbco:hasDfXCriterion* and *mbco:hasDfXIndicator*, creating a structure for computable multi-criteria evaluation as envisioned by [1].

No datatype properties are introduced yet to keep the core ontology lightweight and easily extensible. Quantitative or logical DfX assessment values (e.g., costs, embodied carbon, commonality scores, constraint checks, etc.) can be added later either through task-specific extension modules or external vocabularies. A total of 9 classes and 9 object properties comprise the final core structure of MBCO, as illustrated in Figure 2.

5 Case Study – Creating Instances and Evaluation

To illustrate how MBCO can be used in practice, we model a minimal modular building design (Figure 3) consisting of two volumetric modules and eight planar modules: six wall panels (three solid walls and three opening walls) and two slabs. The goal is to show (i) how a designer can quickly create a consistent instance graph using the MBCO relationships, and (ii) how the same instance data can be evaluated using lightweight, executable checks and computations. The capabilities of CQ1 through CQ6 are utilised for realistic design applications. In this case study, *SolidWall*, *OpeningWall*, *Slab*, *BathroomPod*, and *KitchenPod* are represented as module types (instances of *mbco:ModuleType*). Module properties were attached using IFC-to-LBD-style [26] *props:* predicates so that wall and slab dimensions could be queried directly in a manner consistent with IFC-derived linked data. Additionally, a total of nine connections were instantiated: five of type-1 between volumetric modules and walls, and four of type-2 between walls. The graph thus becomes capable of reasoning over interactions between the modules along with the decomposition structure. The resulting instance graph includes 28 named nodes and 120 RDF triples (i.e. relationships), a part of which is illustrated in Figure 4.

Ontology evaluation ensures logical consistency and testing for fitness for the purpose. In this study, automated consistency checking is done using Protégé HermiT reasoner due to its speed and ease of implementation. It showed no error, confirming the consistency of the ontology. Further, basic task-based evaluation checks are conducted as follows: (i) a lower-level commonality [27] score is computed using the dimensions, types, and connections of planar modules retrieved using SPARQL queries on the populated RDF graph, (ii) manufacturability checks are done on all wall panels to ensure they do not exceed a production line maximum allowable length (set to 8m), and (iii) constructability checks are performed using gross weight of volumetric modules against a crane lifting capacity (set to 4500kg), using SHACL shapes for reasoning.

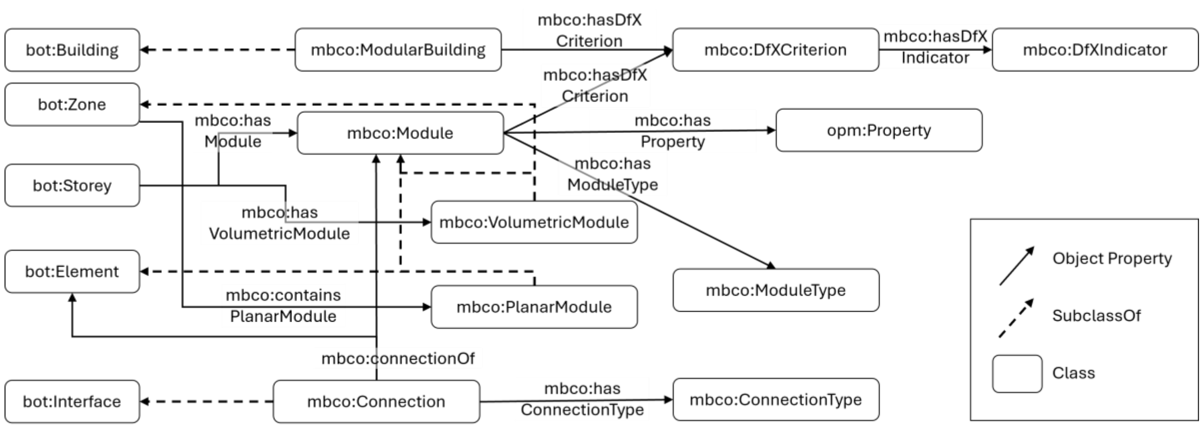

Figure 2: The Modular Building Component Ontology (MBCO)

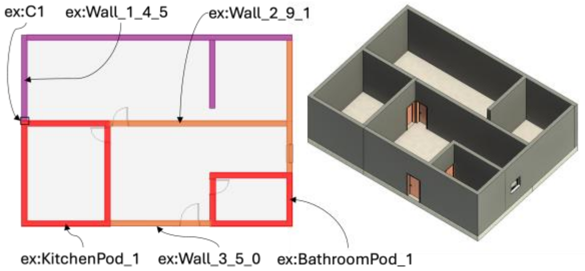

Figure 3. Simple Modular Building Example Case

The computed commonality value of each element and the SHACL reasoning results are populated back into the instance graph using *DfXCriterion* and *DfXIndicator* instances. The SHACL reasoning checks both missing values and values exceeding capacity. Two walls (13.3m and 9.1m) are flagged as being too long against the line limit of 8m. The kitchen pod is flagged as being too heavy against the crane lifting capacity. This demonstrates how MBCO can support the workflow from instance creation through to executable manufacturability and constructability checking and graph-native commonality evaluation.

6 Conclusion

This paper presented the Modular Building Component Ontology (MBCO) as a core, lightweight, interoperable semantic foundation for representing modular buildings in early-stage design. By positioning MBCO as a middle "core" layer aligned to established building ontologies (notably BOT), the ontology captures essential modular concepts like volumetric and planar modules, their containment relationships, and realised interfaces while remaining extensible to downstream task ontologies and external datasets. The case study demonstrated how a simple modular plan can be instantiated efficiently and then evaluated using executable mechanisms: a SPARQL-based commonality metric to quantify reuse across planar modules, and SHACL rules to validate manufacturability and constructability constraints such as production-line length and crane lifting weight limits for various modules. Together, these examples show how MBCO supports a practical workflow from structured representation to computable assessment without embedding project-specific constraints into the core schema. MBCO therefore provides a reusable bridge between modular building semantics and DfX-style decision support, enabling more consistent data integration, automated checking, and possibility for graph-based optimisation. Future work will expand the type catalogues and connection semantics, introduce optional lower-level extension ontologies for performance attributes, and validate the approach on richer industrial case studies integrating BIM/IFC pipelines and multi-objective design optimisation. The data from this research is available at https://github.com/khansalik/MBCO.

References

[1] S.-K. Lee, M.-S. Jeong, T.-Y. Cho, B.-Y. Chung, and J.-H. Yu. Ontology-Based Framework for Context-Aware DfX Criteria Inference Using BIM. *Journal of Management in Engineering*, 41:04025039, 2025.

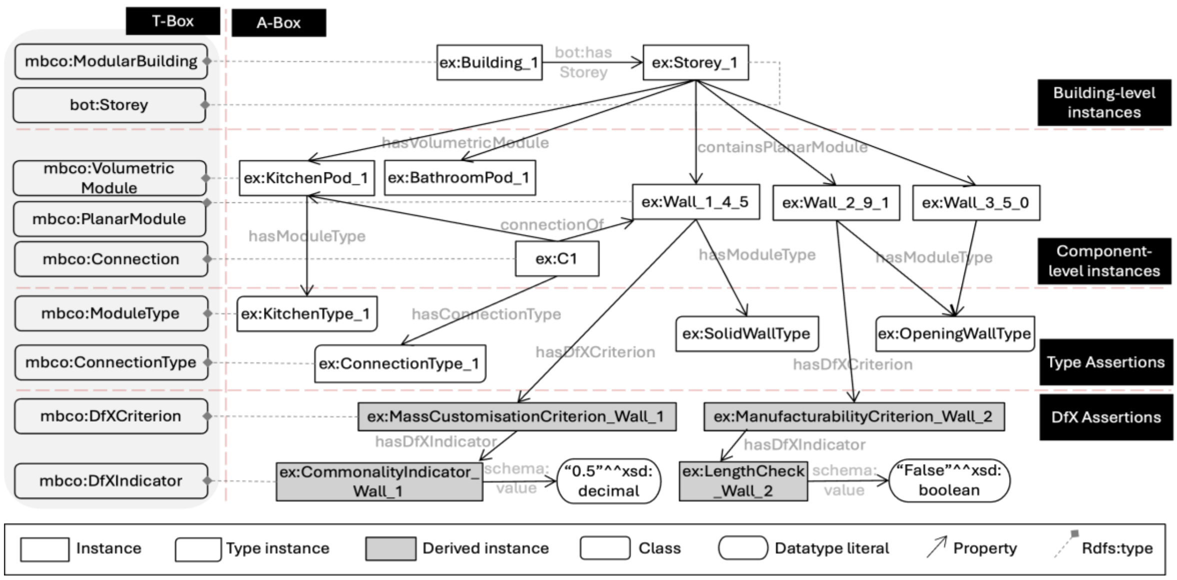

Figure 4: Instance subgraph for the case study populated in accordance with MBCO

[2] P. Ghannad and Y.-C. Lee. Optimizing Modularization of Residential Housing Designs for Rapid Postdisaster Mass Production of Housing. *Journal of Construction Engineering and Management*, 149:04023046, 2023.
[3] T. Salama. Configuration of Hybrid Modular Construction for Residential Buildings. *IJIMT*, pages 106–112, 2017.
[4] A.K. Agrawal, Y. Zou, L. Chen, H. Jin, and M.A. Abdelmegid. An Ontology for Representing Crawler Crane Operational Space Requirement on Semantic Web. In *32nd Annual Conference of the International Group for Lean Construction (IGLC 32)*, pages 648–659, Auckland, New Zealand, 2024.
[5] P. Pauwels, S. Zhang, and Y.-C. Lee. Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73:145–165, 2017.
[6] V.J.L. Gan. BIM-based graph data model for automatic generative design of modular buildings. *Automation in Construction*, 134:104062, 2022.
[7] M.H. Rasmussen, M. Lefrançois, G.F. Schneider, and P. Pauwels. BOT: The building topology ontology of the W3C linked building data group. *Semantic Web*, 12:143–161, 2021.
[8] M.H. Rasmussen, M. Lefrancois, 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*, 2018.
[9] C. Widanage and K.P. Kim. Integrating Design for Manufacture and Assembly (DfMA) with BIM for infrastructure. *Automation in Construction*, 167:105705, 2024.
[10] P. Jensen. *Configuration of Platform Architectures in Construction*. Doctoral thesis, Luleå tekniska universitet, 2014.
[11] A. Wagner, M. Bonduel, P. Pauwels, and R. Uwe. Relating geometry descriptions to its derivatives on the web. In *Proceedings of the conference*, pages 304–313, 2019.
[12] J. Beetz and B. de Vries. Building product catalogues on the Semantic Web. In *Managing IT in Construction/Managing Construction for Tomorrow*, 2009.
[13] S.-K. Lee, K.-R. Kim, and J.-H. Yu. BIM and ontology-based approach for building cost estimation. *Automation in Construction*, 41:96–105, 2014.
[14] S. Bakhshi, M.R. Chenaghlou, F. Pour Rahimian, D.J. Edwards, and N. Dawood. Integrated BIM and DfMA parametric and algorithmic design based collaboration for supporting client engagement within offsite construction. *Automation in Construction*, 133:104015, 2022.
[15] J. Luo, P. Zadeh, and S. Staub-French. Ontology-Based Design Features for Representing Constructability in Architectural Design: Toward BIM in Off-Site Construction. *Journal of Construction Engineering and Management*, 150:05024010, 2024.
[16] J. Cao and D. Hall. Ontology-based Product Configuration for Modular Buildings. In *International Symposium on Automation and Robotics in Construction (ISARC) Proceedings*, pages 171–176, 2020.
[17] M.H. Rasmussen, C.A. Hviid, J. Karlshøj, and M. Bonduel. Managing space requirements of new buildings using linked building data technologies. In J. Karlshøj and R. Scherer, editors, *eWork and eBusiness in Architecture, Engineering and Construction*, pages 399–406. CRC Press, 2018.
[18] P. Pauwels and W. Terkaj. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63:100–133, 2016.
[19] J. Cao, E. Vakaj, R.K. Soman, and D.M. Hall. Ontology-based manufacturability analysis automation for industrialized construction. *Automation in Construction*, 139:104277, 2022.
[20] B. Qi and A. Costin. BIM and Ontology-Based DfMA Framework for Prefabricated Component. *Buildings*, 13:394, 2023.
[21] M. Niknam and S. Karshenas. A shared ontology approach to semantic representation of BIM data. *Automation in Construction*, 80:22–36, 2017.
[22] N.F. Noy and D.L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. 2001.
[23] I.J. Ramaji and A.M. Memari. Product Architecture Model for Multistory Modular Buildings. *Journal of Construction Engineering and Management*, 142:04016047, 2016.
[24] M. Lawson, R. Ogden, and C. Goodier. *Design in Modular Construction*. CRC Press, 2014.
[25] buildingSMART International. Title of the web content: buildingSMART Data Dictionary. On-line: https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/, Accessed: 31/01/2026.
[26] 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 (LDAC 2018)*, 2018.
[27] S.A. Khan, Y. Zou, M. Liarokapis, and M. Kieu. On Component Commonality in Prefabricated Building Design. In *Proc. Int. Workshop Intell. Comput. Eng., EG-ICE, European Group for Intelligent Computing in Engineering (EG-ICE)*, pages 590–599, 2024.