The semantic link between domain-based BIM models

Wojciech Teclaw [1,2], Mads H. Rasmussen [3], Nathalie Labonnote[1] , Jyrki Oraskari [4] *and* Eilif *Hjelseth*[2]

[1] *SINTEF, Trondheim, Norway*
[2] *Norwegian University of Science and Technology, Trondheim, Norway*
[3] *NIRAS A/S, Allerød, Denmark*
[4] *RWTH Aachen University, Aachen, Germany*

Abstract

Over the past few years, the construction industry has undergone technological advancements to improve efficiency and productivity. One of the latest innovations is using semantic web technologies to address interoperability issues and achieve machine interpretability of data. Despite several implementations of Industry Foundation Classes (IFC) to graph model converters, there has been no analysis of the semantic linkages between duplicated elements. This study aims to fill this gap by providing a semantic framework for linking elements in the graph representations of IFC models. This is achieved by reviewing commonly used ontologies, IFC to semantic technology converters, and using the owl:sameAs predicate. The study presents a methodology for generating additional links between duplicated elements in IFC model graph representations using selected geometrical features to address interoperability issues. The methodology is tested on domain-based IFC models and efficiently links models into a federated source of information about interdisciplinary Building Information Modelling (BIM) models. The study's findings are expected to enhance the interoperability and semantic capabilities of BIM models, promoting collaboration and improving the efficiency of the construction industry.

Keywords

Linked Building Data, BIM, IFC-LBD, Semantics, Ontologies, LBD, IFC

1. Introduction

1.1. Context

Over the years, the construction industry faced a few significant technological revolutions. With the advent of CAD (Computer Aided Design), the construction industry was able to streamline design processes and improve collaboration among stakeholders. However, the need for more sophisticated data management systems arose with the increasing complexity of building projects. As early as the late 1980s, researchers recognised the need to improve data exchange in the building and construction sector. In 1995, the International Alliance for Interoperability (IAI) was established to enhance interoperability and productivity in the construction industry. One of the key initiatives of the IAI was the development of the Industry Foundation Classes (IFC), a data model for the exchange of building information. The IAI published the first release of IFC in 1997, marking a significant milestone in the history of Building Information Modelling (BIM).

Nevertheless, facing the pressing issue of excessive CO2 emissions, the construction industry has turned towards developing a new concept known as Digital Twin. This concept's primary objective is to provide a comprehensive and structured representation of building information and equip the industry with autonomous tools capable of controlling, adjusting, and calibrating a construction object throughout its lifecycle, encompassing the construction and operation phases [1]–[3]. The

---

Proceedings Name, Month XX–XX, YYYY, City, Country

EMAIL: wojciech.teclaw@sintef.no (A.1); mhra@niras.com (A.2); nathalie.labonnote@sintef.no (A.3); jyrki.oraskari@dc.rwth-aachen.de (A.4); eilif.hjelseth@ntnu.no (A.5)
ORCID: 0000-0002-3126-0281 (A. 1); 0000-0002-6519-7057 (A. 2); 0000-0001-6790-4109 (A. 3) 0000-0002-4723-3878 (A. 4)

CEUR Workshop Proceedings (CEUR-WS.org)

implementation of digital twin (DT) technology represents a paradigm shift in the construction industry, providing a more efficient and sustainable approach to building management and operation. By incorporating real-time data and advanced analytics, DT technology offers a unique opportunity to optimise building performance, reduce waste, and enhance stakeholder comfort [1], [4]. Despite the numerous benefits of the DT concept, it has singular challenges. DT relies on well-structured, organised, and machine-readable data as input [3]. Therefore, in recent years, multiple research initiatives have focused on applying semantic web technologies as a methodology able to integrate diverse sources of information into a human and machine-readable form. Using IFC models, converting them to Resource Description Framework (RDF) (e.g. in JSON-LD or Turtle) and publishing them as Linked Data (LD) is a natural way to reuse already defined data.[2], [5], [6]. The federated BIM model contains multiple independent models expressing knowledge about certain domains. This approach is aligned with the ISO 19650 standard, being a core of BIM methodology, providing a common interpretability of stand-alone models. However, such an approach causes some knowledge duplication. For example, IfcSpaces and IfcBuildingStoreyElement are exported from every single BIM model containing different GUID (Global Unique Identification Number), even though they represent exactly the same geometrical object representation, an abstract of a room or a level. Parsing such a set of models to LBD format will result in duplicate elements containing exactly the same objects without a direct link or a reference.

1.2. Research questions and paper overview

A methodological framework linking multiple instances of the same space and level geometry occurring in multiple domain BIM models to provide holistic and effortless transfer of information between BIM models. Therefore the following research questions have been formulated:

RQ1: How to efficiently convert a domain-based BIM model to semantic representation?
RQ2: What are the benefits of linking domain-based BIM models in semantic web technologies?

A background is given in Section 2 to provide a common knowledge basis for developing the methodology and the workflow in Section 3. Section 3 also includes the process of selecting representative parameters for element comparison based on the IFC class. Results are presented in Section 4, illustrated by a practical example using a web application. The case study highlights the benefits of the concept and compares the complexity of different SPARQL queries before and after the implementation of the framework. The limitations of the methodology and the opportunities for full-scale applications are discussed in Section 5. Conclusions are given in Section 6.

2. Background

2.1. Ontologies

The development of new ontologies has recently gained prominence as a solution to address interoperability issues and overcome informational deficiencies within various fields. To provide an extendable concrete core of the linked data concept, the W3C Linked Building Data community has identified and established several main ontologies as a standard information exchange format for the community. One of them is ifcOWL, which represents the Industry Foundation Classes data model in Web Ontology Language (OWL) and includes information related to geometry, property sets, elements, and their relationships [7]. In other words, it is an expression of an IFC file in a semantic web format. However, the rigid structure and high complexity of relations of ifcOWL have been found in some studies as inflexible and hard to extend [8], [9].

To address these limitations and provide a flexible and expandable structure, the W3C has defined the Building Topology Ontology (BOT) based on the work of Rasmussen M.H. [10]. BOT is a substantially simplified alternative to ifcOWL, as it focuses on the description of connections between zones, spaces, and building elements. Using BOT results in a more user-friendly experience when querying using SPARQL and reduced graph complexity and size [6].The simplified BOT results in some limitations in comparison to the ifcOWL structure. One of the limitations is that BOT does not provide information on the connectivity of MEP systems. Therefore Flow System Ontology (FSO) was proposed to address this limitation. The ontology comprehensively describes MEP components, their relations, fluid flows, systems, and subsystems. It does not contain information about a link with floors or spaces, which is why understanding the component's context and location requires the usage of alignments such as BOT [11].

The last group of ontologies is an informational supplement to characterising building components using a semantic framework. The PRODUCT ontology enriches a structure's geometric and relational description and correlates features with their physical instantiation as a PRODUCT. It may present an element type of assembly suggested during the design phase, as demonstrated in selected studies [12].

The final PROPS ontology provides a comprehensive set of parameters that describe various aspects of building components. The ontology associates these parameters with a particular PRODUCT or element, enhancing the representation of building components in the BIM model [10]–[12].

2.2. Converters

The researchers and industry professionals community have developed multiple publicly accessible and free tools for converting BIM models. Most focus on converting IFC files to RDF Abox graphs, providing different approaches. Depending upon the technology, users select a tool best for their issue and categorise it, splitting it into server-side and client-side technologies.

The first group gathering solution for server-side tools is based on backend programming languages, which require a dockerised version of an application or a compatible version of a language installed on a local machine. The first tool is IFCtoRDF converting an IFC file to an RDF graph providing a user with an output file using ifcOWL ontology [13]. An alternative was proposed by the IFCtoLBD tool, designed to convert IFC building models into JSON-LD output files by utilising the BOT, PRODUCT, and PROPS ontologies. This tool extracts relevant information from IFC building models and transforms it into simple and effective Abox RDF graphs suitable for use in Linked Data applications [5], [14]. These tools' main advantage is easy integration with applications using API, easy scalability, and the ability to handle large files, which for a client-side browser might be challenging. On the other hand, they require a sufficient amount of computing power and consume additional resources of used infrastructure.

Therefore the second group of converting software are solutions used on the client-side, working as frontend. They might be executed in the browser and use only the client's computational power. The biggest advantage of this technology is that it does not require a server-side to run. Unfortunately, the number of available tools is limited to one. The IFC-LBD converter is an open-source Node Package manager (NPM) package with the configuration output JSON-LD file option to include four different ontologies: BOT, FSO, PRODUCT, and PROPS [15].

2.3. Alignment – owl:sameAs

The studies examining the usage of the owl:sameAs predicate are limited in number. As defined in the OWL ontology documentation, this predicate links two individual URI instances that refer to the same entity. The does not limit the usage of the predicate for Abox or Tbox only. Nevertheless, it is commonly utilised to establish a mapping between different ontologies, linking two equal Tbox classes [16]. Some studies caution against its use, as it assumes that two corresponding object instances are identical in every aspect, including all their parameters. This could result in misleading conclusions if not satisfied [8]. Meanwhile, studies from various domains, such as computer science, have emphasised the importance of conducting thorough data analysis to validate the use of the owl:sameAs predicate between two elements [17]. Unfortunately, available documentation does not unambiguously classify the boundary conditions and clearly define implementation situations. They rely upon knowledge and judgment depending on the use case.

3. Methodology

3.1. IFC models conversion

The tools discussed in section 2.2 provide various output results and formats, all of which are related to representing semantic web graphs. The following criteria were established for selecting the most appropriate tool for this study.

The main criteria for selecting the most convenient technology were implementation effort and tool flexibility. The IFC-LBD solution was found to perfectly meet the requirements of the study, providing users with versatile configuration options, and simplifying the graph structure compared to the ifcOWL assertions generated by IFCtoRDF [5].

The IFC-LBD proposed by Rasmussen M.H. et al. [15] converter is designed as an integrated extension of the web-ifc library, used for parsing IFC data into the library, thus requiring only one programming technology (JavaScript or TypeScript) instead of using different dockerised services and integrating them using a microservices architecture approach.

After converting an IFC file to JSON-LD format, the knowledge was serialised in the triple store (database), as presented in Figure 1.

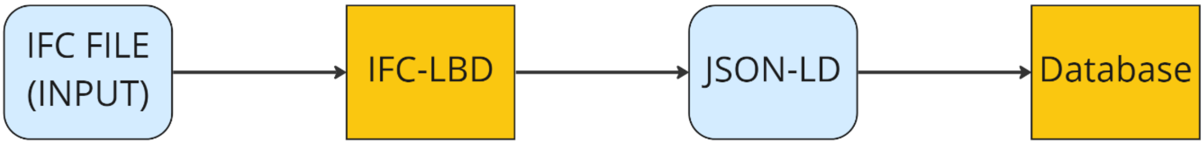


Figure 1: A file conversion workflow

When exporting an IFC file from a native model, each IFC model element is assigned a unique GUID identifier to differentiate it from other components. Furthermore, for generated structures in the model or after edits to existing models, it is important to note that the GUIDs assigned to the elements may not remain the same when the model is exported consecutively.

Therefore the converter offers the option to customise the project-specific namespace, which, combined with the GUID, provides a unique URI address for each instance. This result could also be achieved using other available tools [14]. The main barrier of other converters is the need for additional postprocessing work related to geometry representation, which would allow efficient and flexible reflection of an object.

3.2. Elements characteristic features

3.2.1. IfcSpace similarities

The similarities between spaces in the following framework require exactly the same matching between model elements' selected parameters. The first step is streaming all space geometry from an IFC model and their conversion into the universal BufferGeometry object provided by the three.js library. The such expression enables calculating the first key parameter: volume, based on an object's geometry, not relying on properties. In the next step, the bounding sphere of the geometry is calculated using built-in functionalities, enabling the usage of a centre point and radius, as presented in Figure 2.

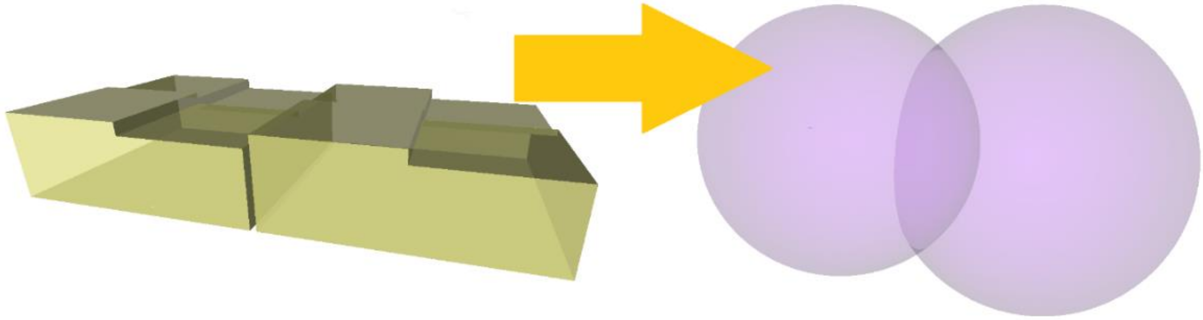


Figure 2: Conversion of space geometry to bounding spheres

The generated properties provide a string input for a context string representing a space. The context string formula concatenates the following values:

- Bounding sphere centre point – X coordinate
- Bounding sphere centre point – Y coordinate
- Bounding sphere centre point – Z coordinate
- Bounding sphere radius
- The space volume
- The number of vertices creates a space geometrical representation.

3.2.2. IfcBuildingStorey similarities

A comparison of IfcBuildingStorey elements required significantly less effort than in the case of IfcSpaces. A level is a plane always parallel to the XY plane of a model. Therefore researchers proposed generating the context string representation based on two basic parameters - name and level elevation.

3.3. Hashing

The process of aligning and comparing context strings is facilitated using hash values generated by a GUID representation. It is important to note that the usage of GUID version 5 is central to this process. Unlike GUID version 4, which generates a randomly generated 2^{128} value, GUID version 5 generates a contextual hash based on the input parameters of the namespace and input string [18]. This returns the same hash value for the same input string, facilitating a simple and efficient comparison process. For example, if the algorithm inputs context strings representing identical geometries, it will always produce the same GUID number, thereby simplifying the comparison process.

3.4. Elements comparison

The proposed methodology requires a comparing IFC model element focusing on two building-specific classes representing a location. IfcSpace and IfcBuildingStorey represent an abstraction of a building's physical room element and level. The BOT ontology used bot:containsElement and bot:hasElement to express bot:Element relationship with IfcSpace or IfcBuildingStorey, respectively [10]. Each model space or level instance might contain information about hundreds of elements spread across different domains. Therefore, the geometrical hashing values were calculated to avoid a heavily iterative process. Because of that ifcSpaces and IfcBuildingStoreys are compared by hashed values only, which results in a well-performing algorithm.

The last step of the proposed methodology is adding newly discovered triples to the database. None of the tools discussed in section 2.2 provides a unified approach for linking multiple files and adding additional predicates not directly specified in the IFC file. Therefore, in the following part of the section, based on the usage of base properties and geometric characteristics for linking the same objects occurring in IFC models was proposed by adding the owl:sameAs predicate bidirectionally, providing the knowledge that two elements occurring with two models are exactly the same. The operation of discovery of new connections is made automatically by a tool proposed in section 4.4.

4. Results

4.1. Case study models description

The following section presents the implementation of the methodology. Using Autodesk Revit, three IFC models were generated for a given building, respectively, including architectural, plumbing and ventilation domains. They are presented in Figure 3:

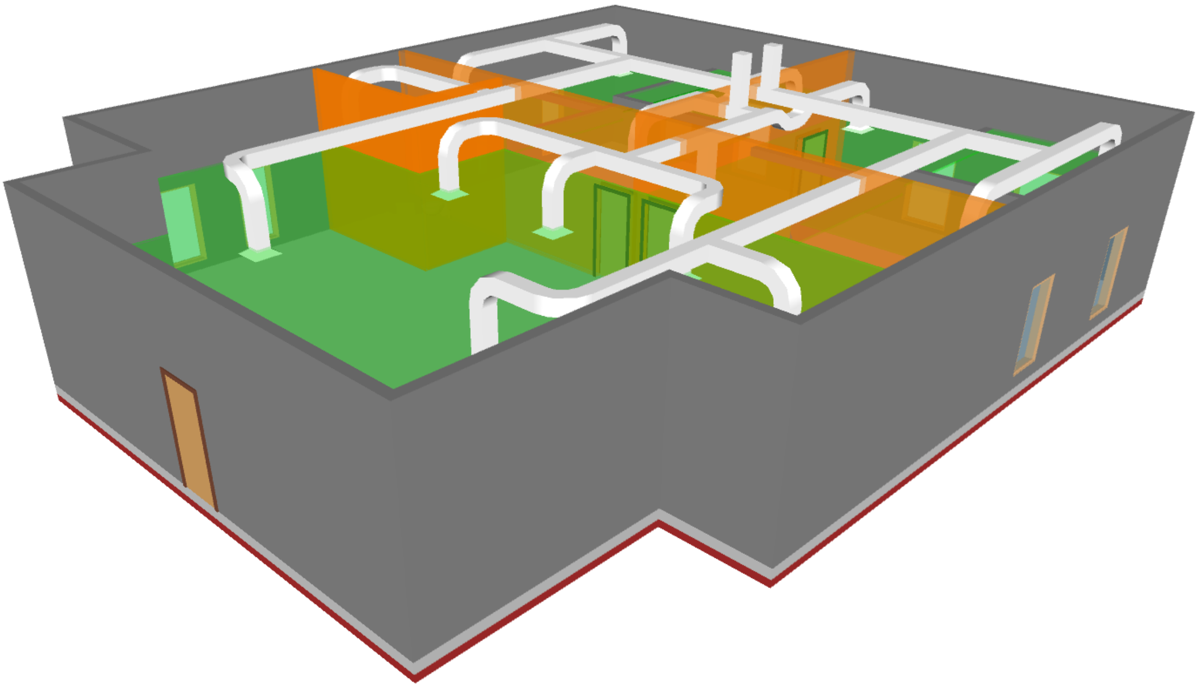


Figure 3: Federated model of three IFC models

All models are generated from Autodesk Revit to IFC separately using a standard IFC 2x3 Coordination View 2.0 skimmer, one of the standardised approaches of ISO 19650. They contain the following elements:

Architecture:

- Ceilings (IfcCovering)
- Doors (IfcDoor)
- Floors (IfcSlab)
- Level (IfcBuildingStorey)
- Rooms representation (IfcSpace)
- Walls (IfcWall)
- Windows (IfcWindow)

Pipiling model:

- Pipes (IfcFlowSegment)
- Pipe fittings (IfcFlowFitting)
- Rooms representation (IfcSpace)
- Terminals (IfcTerminal)
- Systems:
    - Wastewater system
    - Coldwater supply

Ventilation model:

- Ducts (IfcFlowSegment)
- Duct fittings (IfcFlowFitting)
- Rooms representation (IfcSpace)
- AirTerminals (IfcTerminal)
- Systems:
    - Return Air
    - Supply air

4.2. Knowledge integration

In accordance with the methodology outlined in Section 3, the IFC-LBD was utilised to convert each individual IFC model into JSON-LD format [15]. The BOT, FSO, PROP ontologies and custom namespaces were utilised to maximise the range of query options [5], [10], [11]. The resulting output was transformed into N-Quads and stored in the triple store. The conversion process was executed asynchronously across multiple IFC files. Each file took just a few milliseconds to process, resulting in only a minimal number of unused properties or relations.

```
PREFIX bot:<https://w3id.org/bot#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>

CONSTRUCT {
  ?subject ?predicate ?object .
}
WHERE {
  ?subject ?predicate ?object .
  FILTER(?object = bot:Space || ?object = bot:Storey || ?predicate = owl:sameAs)
}
```

Listing 1: SPARQL query returning all bot:Space elements and owl:sameAs relations

The following step involved the analysis of geometrical similarities based on the hashed values of the geometry representation context strings. The geometry representation was hashed using a limited set of parameters, avoiding the need for computationally expensive CSG (Constructive Solid Geometry) intersections. This comparison process of the three models increased the number of graph elements by the additional 30 owl:SameAs predicates, bringing the total number of graph elements from 6086 to 6116. The impact of these additional predicates on the triple store can be retrieved using the SPARQL query presented in Listing 1. This query retrieves all elements that are bot:Space or bot:Storey, as well as all elements whose nodes are connected by owl:sameAs.

The query result visualising the linking process is presented in Figure 4, where five logical groups might be distinguished. Four groups express the information about links referring to the geometrical room representation. Each element is connected using a bidirectional owl:sameAs predicate representing the equivalence between nodes.

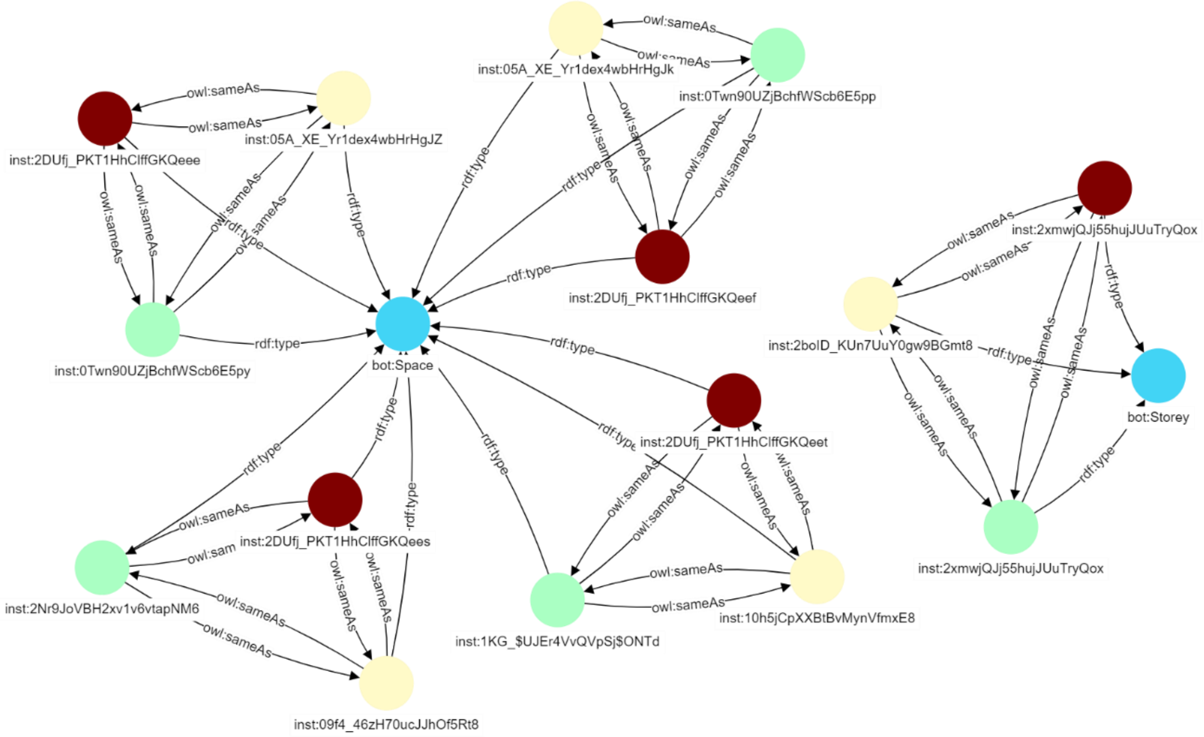


Figure 4: The bot:Space and bot:Storey elements and their owl:sameAs relations

4.3. Usage owl:sameAs

The usage of additional predicates might be used differently depending upon a need and expected result. The selected demonstrates the practical usage of the framework by reaching elements stored in various IFC models. One of the use cases is querying the names of all mechanical systems impacting the space. In the example, an architectural model space representing a toilet was selected. The SPARQL query presented in Listing 2 returns all systems names supplying or returning fluid to a particular element.

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX bot:<https://w3id.org/bot#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX fso:<https://w3id.org/fso#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?SELECTED_SPACE ?SYSTEM_NAME
WHERE
{
    BIND(http://www.sample.org/architecture/09f4_46zH70ucJJhOf5Rt8 as ?SELECTED_SPACE)
    {
        ?SELECTED_SPACE      ?predicate            ?object .
        ?object       rdf:type      fso:Terminal .
        ?systemElement fso:hasComponent    ?object .
        ?systemElement rdfs:label   ?SYSTEM_NAME   .
    }
    UNION
    {
        ?SELECTED_SPACE      owl:sameAs    ?theSameSpaceObject .
        ?theSameSpaceObject  ?predicate            ?object .
        ?object       rdf:type      fso:Terminal .
        ?systemElement fso:hasComponent    ?object .
        ?systemElement rdfs:label   ?SYSTEM_NAME   .
    }
}
```

Listing 2: SPARQL query returning MEP system names in selected space

Another example is a query returning all components placed on a selected level. In the example, the level from the heating and cooling model was selected and based on the SPARQL query, it returns 128 bot:Elements presented in Listing 3.

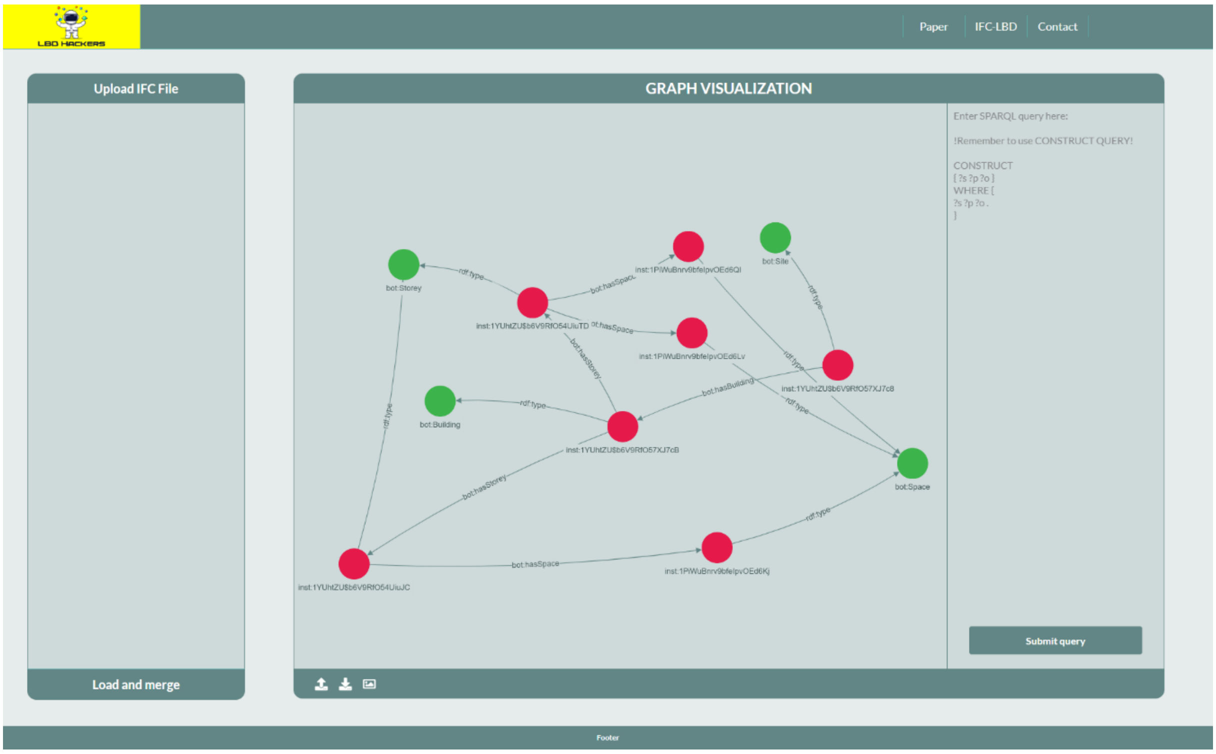
```
PREFIX bot:<https://w3id.org/bot#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?element
WHERE {
    BIND (<http://www.sample.org/piping/2xmwjQJj55hujJUuTryQox> AS ?storey )
    {
        {
            ?storey bot:hasElement  ?element .
        }
        UNION
        {
            ?storey bot:hasSpace    ?space .
            ?space bot:containsElement     ?element .
       }
    }
    UNION
    {
        ?storey owl:sameAs ?otherStorey .
        {
            ?otherStorey bot:hasElement    ?element .
        }
        UNION
        {
            ?otherStorey bot:hasSpace      ?space .
            ?space bot:containsElement     ?element .
       }
    }
    ?element rdf:type ?type
    FILTER (?type = bot:Element)
}
```

Listing 3: SPARQL query returning all elements from Storey and its equivalents

4.4. Web application

As a result of this research, the web application was developed, providing users with the methodological framework in a feasible form, publically available at: https://wojciechteclaw.github.io/LBD-Converter-Online/. It offers the option to upload IFC files, customise the parsing options, merge models, query the database and display the graph. Moreover, the results of file merging might be downloaded as an N-Quads file and uploaded to any other software. The solution was developed using Typescript language, React framework, web-ifc and ifc-lbd libraries. The application interface is presented in Figure 5.



```
Enter SPARQL query here:

!Remember to use CONSTRUCT QUERY!

CONSTRUCT
{ ?s ?p ?o }
WHERE {
?s ?p ?o .
}
```

Figure 5: LBD-Visualiser application interface overview

5. Discussion

The following paper aims to develop a methodological framework for semantic enrichment between BIM model similarities. The demonstration showed that such an approach might benefit Data Scientists for new data and relations discovery. It might ensure improved collaboration and interoperability between stakeholders, which can benefit from the solution in multiple ways. The first is the direct exchange of parameters. While modelling a construction object, different domains introduce required parameters, such as the number of occupants in a room, heating gains, or room type. Using the proposed approach, the knowledge might be easily accessible and editable between various models (RQ2).

The framework might be applied using any other research converter because it relies on the raw spaces, raw geometry and a level of the basic properties which are always generated while exporting a model [13]–[15]. The main benefit of the methodology is its speed because it does not have to perform any CSG operations, which highly consumes computational power (RQ1). During the development, an attempt to use the CSG engine was made using dedicated workers. Due to the number of iterations, it could work with relatively small IFC files. Implementation of the CSG approach would make the concept much more flexible. It would not require 100% geometry similarity but could be 95% of volume coverage, allowing domain-specific modellers for flexibility. Such a case would require an extension to an existing ontology by a new predicate describing that two space elements are not exactly the same, but they represent one physical element. For example, the predicate could be named bot:geometricallyEquivalent and useful to the geometrical representation of objects between various domains.

The owl:sameAs predicate ensures the data between duplicated elements representing buildings' location is easily accessible, benefiting all stakeholders of the construction process. Based on provided web application, researchers and AEC industry professionals might use the concept for knowledge discovery and explore the usage of the methodology.

6. Conclusion and further work

Despite the differences in modelling approaches and domain-specific IFC models, the paper demonstrated the successful usage of the owl:sameAs predicate to enrich the information between multiple IFC models. The presented methodology proposes a solution for handling differences in GUIDs of the same elements stored in various domain-specific IFC files and demonstrates it in a web application. The concept might provide a significant improvement for the initial data mining for the purpose of digital twin development because of the ability of an interoperable approach between domains. The framework might be successfully implemented from design to operation because it does not interfere with existing workflows and project standards.

The further work of researchers will focus on the extension of the concept by proposing a complementary methodology linking the interdisciplinary MEP elements, which connectors do not have intermodel relationships. A combination of these two approaches will provide a close system which will create a basis for the BIM to Digital Twin approach, enhancing the multidisciplinary interoperability.

7. References

[1] S. Hu, J. Wang, C. Hoare, Y. Li, P. Pauwels, and J. O'Donnell, "Building energy performance assessment using linked data and cross-domain semantic reasoning," *Autom. Constr.*, vol. 124, no. February, p. 103580, 2021, doi: 10.1016/j.autcon.2021.103580.

[2] P. Pauwels, S. Zhang, and Y. C. Lee, "Semantic web technologies in AEC industry: A literature overview," *Autom. Constr.*, vol. 73, pp. 145–165, 2017, doi: 10.1016/j.autcon.2016.10.003.

[3] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic Construction Digital Twin: Directions for future research," *Autom. Constr.*, vol. 114, no. November 2019, p. 103179, 2020, doi: 10.1016/j.autcon.2020.103179.

[4] A. Donkers, B. de Vries, and D. Yang, "Knowledge Discovery Approach to Understand Occupant Experience in Cross-Domain Semantic Digital Twins," *CEUR Workshop Proc.*, vol. 3213, pp. 77–86, 2022.

[5] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, and R. Klein, "The IFC to linked building data converter - Current status," *CEUR Workshop Proc.*, vol. 2159, no. June, pp. 34–43, 2018.

[6] P. Pauwels, S. Törmä, J. Beetz, M. Weise, and T. Liebich, "Linked Data in Architecture and Construction," *Autom. Constr.*, vol. 57, pp. 175–177, 2015, doi: 10.1016/j.autcon.2015.06.007.

[7] W. Terkaj and A. Šojić, "Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology," *Autom. Constr.*, vol. 57, pp. 188–201, 2015, doi: 10.1016/j.autcon.2015.04.010.

[8] F. Beck, J. Abualdenien, and A. Borrmann, "An evaluation of the strict meaning of owl:sameAs in the field of BIM GIS Integration," *CEUR Workshop Proc.*, vol. 3081, pp. 154–165, 2021.

[9] J. Werbrouck, P. Pauwels, J. Beetz, and L. van Berlo, "Towards a decentralised common data environment using linked building data and the solid ecosystem," *Adv. ICT Des. Constr. Manag. Archit. Eng. Constr. Oper. Proc. 36th CIB W78 2019 Conf.*, no. September, pp. 113–123, 2019.

[10] M. H. Rasmussen, M. Lefrançois, G. F. Schneider, and P. Pauwels, "BOT: The building topology ontology of the W3C linked building data group," *Semant. Web*, vol. 12, no. 1, pp. 143–161, 2020, doi: 10.3233/SW-200385.

[11] V. Kukkonen, A. Kücükavci, M. Seidenschnur, M. H. Rasmussen, K. M. Smith, and C. A. Hviid, "An ontology to support flow system descriptions from design to operation of buildings," *Autom. Constr.*, vol. 134, no. February, p. 104067, 2022, doi: 10.1016/j.autcon.2021.104067.

[12] A. Wagner, W. Sprenger, C. Maurer, T. E. Kuhn, and U. Rüppel, "Building product ontology: Core ontology for Linked Building Product Data," *Autom. Constr.*, vol. 133, no. October 2021, p. 103927, 2022, doi: 10.1016/j.autcon.2021.103927.

[13] P. Pauwels, "pipauwel/IFCtoRDF." 2020, Accessed: Feb. 09, 2023. [Online]. Available: https://github.com/pipauwel/IFCtoRDF.

[14] J. Oraskari *et al.*, "jyrkioraskari/IFCtoLBD." Zenodo, 2021, doi: 10.5281/zenodo.5772656.

[15] M. H. Rasmussen, "IFC-LBD." 2023, Accessed: Feb. 09, 2023. [Online]. Available: https://github.com/LBD-Hackers/IFC-LBD/.

[16] S. B. F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, and L. A. Stein, "OWL Web Ontology Language Reference, http://www.w3.org/TR/2004/REC-owl-ref-20040210/," *W3C Recommendation*, vol. 2, no. February. p. http ://www.w3.org/TR/owl-ref/, 2004, Accessed: Feb. 09, 2023. [Online]. Available: https://www.w3.org/TR/owl-ref/.

[17] R. Parundekar, C. A. Knoblock, and J. L. Ambite, "Linking and Building Ontologies of Linked Data," Accessed: Feb. 09, 2023. [Online]. Available: http://www.ncbi.nlm.nih.gov/entrez/query/static/help/.

[18] P. Leach, M. Mealling, and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace," *Request for Comments*, 2005. http://www.itu.int/ITU-T/asn1/uuid.html (accessed Feb. 19, 2023).