A Framework Integrating NLP and Semantic Web to Query IFC Building Information Models with a Prior Ontology Population Process: Information Collection Sheet

**Introduction**

Over the recent years, Building Information Modelling (BIM) serves as one of the most notable technologies for information management in the architecture, engineering and construction (AEC)/facilities management (FM) industry. Using Industry Foundation Class (IFC) data format, BIM data could be exchanged among different construction-related software and tools, which enables information standardization and interoperability for the industry. Since more and more relevant disciplinaries are concerned and integrated into the schema-level or instance-level BIM data, the model becomes structurally complex, which makes it difficult for users to fully understand the entities and hierarchy appeared in an IFC instance file. It poses troubles for end-users to access the information towards BIM data with specific requirements. Currently, there are some specialized query languages (e.g. BIMQL) and architectures (e.g. BIMServer) supporting information retrieval of BIM data. However, utilizing such query languages needs the skills of programming, and understanding the grammars of query languages and the structure of the IFC schema. Obviously, it is not practical to expect that the stakeholders in construction projects could be proficient in these skills and knowledge. Thus, a flexible and straightforward approach to conduct information retrieval of BIM data is needed.

IFCOWL is a newly emerging representation of IFC data based on Ontology Web Language (OWL), and has been acknowledged as an official format by BuildingSMART. Compared with IFC organized in EXPRESS, IFCOWL allows formal knowledge representation and semantic data sharing of building information models taking advantages of the power of semantic web technologies (Pauwels & Terkaj, 2016). BimSPARQL, a domain-specific functional SPARQL extension, is published to query IFCOWL instance files (Zhang et al., 2018). The end-users still need to write SPARQL query sentences on their own to access the required BIM data in RDF graphs.

Based on abovementioned works, this research proposes a framework that integrates natural language processing (NLP) and semantic web technology to query IFC BIM models in OWL representation. It aims to interpret a natural language query, such as “Find the shear walls in the level 2.” and convert it into a computer-readable SPARQL query sentence to acquire the precise answers. Two research questions are asked: (1) how to jointly extract the mentioned concepts in a natural language query and infer their semantic relationship? (2) how to automatically convert the mapping and inference results into SPARQL query sentences? In order to answer the above questions, this research develops a system consisting of (1) Initialization Module (2) Concept Extraction Module (3) Relationship Inference Module (4) Templated-based SPARQL Query Generation Module.

To evaluate the system and further improve its performance, it is necessary to collect and test amounts of end-users’ queries. The reason stems from the fact that end-users have various patterns of expressions of the same request. For instance, when asking walls with a dimension constraint, one might input “Find walls with length less than 5 meters” while the other might input “Select walls of which the length is smaller than 5 m”. To ensure the robustness of the system, a variety of expressions of natural language queries should be tested. Hence, this research invites BIM professionals both from academia and industry to write text-based queries with regards to any information they want to acquire given an IFC model.

**Requirements**

* Each invited BIM practitioner is assigned with an IFC model. The participant writes **20** natural language query sentences with regards to the assigned IFC model.
* The participant is suggested to cover as many as elements, constraints and functional relationships as described in the next section in the 20 sentences.
* The participant is suggested to try to avoid the similar expression as given in the example when he/she writes the 20 query sentences.
* The participant can accord to the Appendix 1 that provides sample query sentences when he/she writes the 20 query sentences.

**Scope**

Currently, the developed system only supports querying the IFC models with a limited scope of objects, constraints and relationships.

**For objects:**

the supported elements that can be queried in IFC model are ***IfcSharedBuildingElement*** and ***IfcSpatialStructureElement***. In details, user can query wall, door, window, footing, slab, column, covering, curtainwall, member, pile, plate, railing, ramp, rampflight, roof, stair, stairflight as building elements, and space, storey as spatial elements. The system also extends to the instance-level querying. Namely, the user can query an instance with its name or put it as constrain (e.g. Select the doors placed in Basic Wall:Exterior - Brick on Block:138157(a wall instance)).

**For constraints:**

the supported constraints that can be queried are ***IfcProperty, IfcPhysicalQuantity****,* ***IfcMaterial (including Material List and Material Layer)****,* ***IfcTypeObject (Type Name and Enum)***, ***Type***. Subject to the data types of some constraints such as property, there are three types of value constraints:

1. Boolean Constraints: For example, “**Select the loadbearing walls**”, the property “Loadbearing” is a Boolean constraint, it can be TRUE or FALSE.
2. Quantitative Constraints: For example, “**Select the wall with length less than 5 meters**”, the property “Length” has Double datatype. The system currently supports the following operators for quantitative constraints: equal, comparative (larger than, less than, not less than, not larger than), maximum and minimum.
3. String Constraints: For example, “**Select the wall with its phase created at new construction**” the property “phase created” has value type of String. Here, the value is “new construction”.

**For functional relationships:**

the system currently supports extra functional relationships that can be considered as constraints as follows:

1. Hascontainedproduct: Relationship between spatial structure element and its contained product.

E.g. “Find the space that contains (hascontainedproduct) Basic Wall:Exterior - Brick on Block:138157”.

b. Haselementcomposition: Provision of an aggregation structure where the element is part of another element representing the composite.

E.g. “Find the stairflights that compose (haselementcomposition) the Stair:Residential - 200mm Max Riser 250mm Tread:151086.”

c. Haselementdecomposition: Provision of an aggregation structure where the element, representing the composite, is decomposed into parts represented by other elements.

E.g. “Find the stair that is decomposed by Railing:1100mm Guard Rail:151166.

d. Hasnextspace: Relationship between a space and a space next to it. They share the same space boundary.

E.g. “Find the spaces next to the room A102.”

e. Hasspaceboundary: Relationship between a space and its boundary elements, which can be physical (e.g. wall, door) or virtual (virtual element)

E.g. “The space has boundary element of Basic Wall:Exterior - Brick on Block:138157”.

f. Hasspatialdecomposition: Spatial hierarchy relationship between project or spatial element ( site, building, storey, space)

E.g. “The floor that is decomposed by the room A102 and A103.

g. Iscontainedin: Relationship between a product and a spatial structure element which contains it.

E.g. “The walls at level 1”.

h. Isplacedin: Relationship between elements like door and window and their placed walls, slabs or other elements

E.g. “The doors placed in Basic Wall:Exterior - Brick on Block:138157.”

**Logic Operators:** support Logic “AND” and “OR”. E.g. Select the wall with length larger than 5 and width less than 3. Overall, the scope of query is shown in Figure 1.

**Other tips:**

* Currently, the system only supports “Select” query, other types such as “Count” are not allowed.
* Currently, the system don’t support a query in form of a question, such as “What, How, Where”.

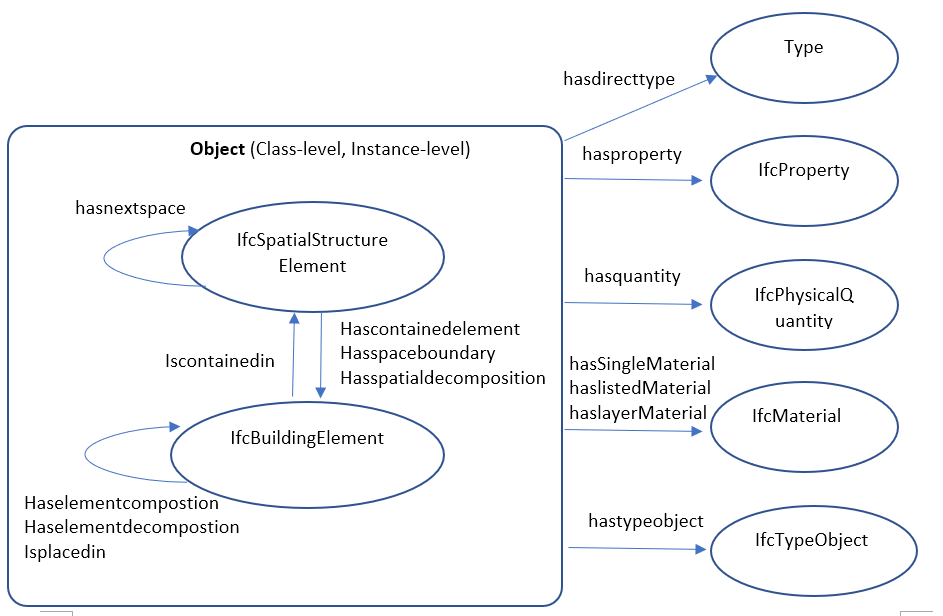


Figure 1 Scope of Query

**Query Sentence Collection**

Please input your 20 sentences in the below slots:



**Appendix 1.**

The existing sample query sentences for reference.

texta1 = "The window with fixed assetaccountingtype."; //window with string constrain of property

texta2 = "The walls at Level 1."; //wall with spatial constrain

texta3 = "The wall with top constraint up to level:roof."; //wall with property string constrain

texta4 = "The space with gsa bim area more than 10"; //space with quantity constrain

texta5 = "The wall with location line of 2"; // wall with int quantity constrain

texta6 = "The wall that has length less than 5"; //wall with quantity constrain

texta7 = "The door that is single\_swing\_right "; //wall with type enum constrain

texta8 = "The wall with type of Basic Wall:Exterior - Brick on Block:130892 "; //wall with type constrain

texta9 = "The window of which the glass pane material is glass "; //window with property string constrain

texta10 = "The wall with material layer of masonry brick not less than 0.5"; //wall with layer material and depth constrain

texta11 = "Select basic wall:party wall - cmu residential unit dimising wall:143239"; //select a instance

texta12 = "The load bearing walls."; // boolean property constrain

texta13 = "The doors placed in the basic wall:party wall - cmu residential unit dimising wall:143239. "; //doors placed in a certain wall instance

texta14 = "The double\_door\_sliding doors in the shear wall at level 2. "; //door with multiple constrains.

texta15 = "The doors in the external wall. ";

texta16 = "The wall that contains metal stud layer."; //wall with material constrain

texta17 = "Select the slabs with height offset from level larger than 5m at the second floor."; //slab with multiple complex constrain.

// texta18 = "Get all walls within the first storey."; //wall with spatial constrain.

texta18 = "Get all walls within level 1."; //wall with spatial constrain.

texta19 = "Get the spaces next to A102."; //The spaces that is next to a certain ifcspace instance.

texta20 = "The walls that belong to Basic Wall:Party Wall - CMU Residential Unit Dimising Wall:128555."; // wall with direct type constrain

texta21 = "The space with gsa bim area not more than 5 cubic meter."; //space with quantity constrain

texta22 = "The space that has boundary element Basic Wall:Party Wall - CMU Residential Unit Dimising Wall:139234."; //space with quantity constrain

texta23 = "The floor where A102 locates."; // the storey with space constrains through hasspatialdecomposition relationship.

texta24 = "The objects that composite Stair:Residential - 200mm Max Riser 250mm Tread:151086"; // find the object that composite a stair instance with relationship: haselementcomposition.

texta25 = "The floor of which area is larger than 45 "; // the floor with property (area) constrain

texta26 = "The footings with material of Concrete - Cast In Situ. "; // the footing with property (material) string constrain.

texta27 = "The beams contained in level 1. "; // beam with storey cosntrain.

texta28 = "The beams with reference level of roof. "; // beam with string property constrain

texta29 = "The external beams with span larger than 4. "; // beam with multiple property constrain.

texta30 = "Get the objects placed in the Basic Wall:Exterior - Brick on Block:138157. "; // find the objects placed in a wall instance.

texta31 = "Find the slabs composing Basic Roof:Live Roof over Wood Joist Flat Roof:184483. "; //find the slab that composite a roof instance

texta32 = "The roof with total area larger than 132. "; //find the roof with property quantity constrain

texta33 = "Select the coverings with depth of plasterboard smaller than 0.02. "; //coverings with material layer constrain

texta34 = "The coverings that have metal stud layer not more than 0.05 meter. "; //coverings with material layer constrain

texta35 = "The coverings at level 2. "; //coverings with storey constrain

texta36 = "The coverings with totalthickness of 0.057. "; //coverings with property quantity constrain

texta37 = "Select the room bounding coverings. "; //coverings with storey constrain

texta38 = "The coverings of which assetaccountingtype is fixed. "; //coverings with property string constrain

texta39 = "The coverings which are not room bounding. "; //coverings with storey constrain

texta40 = "Find the foundation under the condition that its length is no more than 18 meter and volume is less than 5."; //find footings with multiple property constrain.

texta41 = "Select all objects in the roof level."; //find all objects in a certain level

texta42 = "Select all building elements within level 1."; //find all objects in a certain level

texta43 = "Get the spaces surrounding A201"; //find all space instance next to a certain space instance.

texta44 = "Select the 750mm x 2200mm windows"; //find all windows with a typeobject constrain.

texta45 = "Select the single swing left doors"; //find doors with type enum constrain.

texta46 = "Select the doors with operation type of SINGLE\_SWING\_LEFT."; //find doors with type enum constrain.

texta47 = "Select the walls that contain masonry-brick and plasterboard."; //wall with multiple material constrain.

texta48 = "Select the spaces in the level 1."; //wall with multiple material constrain.

texta49 = "Find the external doors."; //door with property boolean constrain

texta50 = "Find the windows of which the frame exterior material is made of sash."; //window with a property string constrain

texta51 = "The floor with elevation higher than 3 meter."; //floor with property quantity constrain

texta52 = "The wall with 2 location lines";// wall with int quantity constrain

texta53= "Select all shear walls";// wall with type enum constraint

texta54 = "The spaces next to B205."; //The spaces that is next to a certain ifcspace instance.

texta55 = "The footings with its length is no more than 18 meter and volume is less than 5."; //find footings with multiple property constrain

texta56 = "Select the elements within T/FDN."; //find all objects in a certain level

texta57 = "Get the places connected with B205."; //find all space instance next to a certain space instance.

texta58 = "The wall with length less than 18 and area larger than 40.";

texta59 = "The wall with length less than 18 and length larger than 15.";

texta60 = "The length of beams at level 1.";

texta61 = "The wall with length at least 15 meter.";

texta62 = "The walls at level 1 and level 2.";

texta63 = "The wall with length less than 18, or area larger than 40.";

texta64 = "The wall of which the length is less than 18, or the area is larger than 40.";

texta65 = "The wall of which the length is less than 18 or the area is larger than 40.";

texta66 = "The windows and doors in the Basic Wall:Exterior - Brick on Block:138157.";

texta67 = "The building elements in the Basic Wall:Exterior - Brick on Block:138157.";

texta68 = "The walls at level 1 or level 2.";

texta69 = "The wall made of Masonry - Brick and its length is larger than 17.";

texta70 = "The wall made of Masonry - Brick or its area is larger than 40.";

texta71 = "The wall with the largest length.";

texta72 = "The wall with the minimum length.";

texta73 = "The level with the highest elevation.";

**Appendix 2.**

If user doesn’t have IFC viewer on hand, he/she is suggested to download XBIM Xplorer . Address: <https://docs.xbim.net/downloads/xbimxplorer.html>