Automation in Construction 95 (2018) 46–63

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

# A SPARQL query engine for binary-formatted IFC building models

Thomas Krijnen[a,*], Jakob Beetz[b]

[a] Eindhoven University of Technology (TU/e), Eindhoven, Netherlands
[b] RWTH Aachen University, Templergraben 55, Aachen 52062, Germany

ARTICLE INFO

*Keywords:*
BIM
IFC
Querying
SPARQL
Performance
HDF5

ABSTRACT

To date, widely implemented and full-featured query languages for building models in their native exchange formats do not exist. While interesting proposals exist for querying Industry Foundation Classes (IFC) models, their functionality is often incomplete and their semantics not precisely defined. With the introduction of the *ifcOWL* ontology as an equivalent to the IFC schema in the Web Ontology Language (OWL), an option to represent such models in RDF (Resource Description Framework, a general information modeling method) is provided, and such models can be queried using SPARQL (SPARQL Protocol and RDF Query Language). The size of data sets in complex building projects, however, renders the use of clear-text encoded RDF infeasible in many cases.

A SPARQL implementation, compatible with *ifcOWL*, is proposed, directly atop a standardized binary serialization format for IFC building models. This novel format is the binary equivalent of traditional IFC serialization formats but with more compact storage and less overhead than the graph serialization in RDF. The format is based on ISO 10303-26 and relies on an open standard for organizing large amounts of data: Hierarchical Data Format version 5 (HDF5). Due to hierarchical partitioning and fixed-length records, only small subsets of the data are read to answer queries, improving efficiency.

A prototypical implementation of the query engine is provided in the Python programming language. In several realistic use cases, the proposed system performs equivalent to or better than the state of the art in SPARQL querying on building models. For large datasets, the proposed storage format results in files that are 2–3 times smaller than the current, most concise, RDF databases while offering a platform-neutral, containerized exchange file.

## 1. Introduction

With the advent of the Building Information Modeling (BIM) paradigm, buildings are exchanged as rich parametric and semantic data models. This development is a significant improvement over exchanging information as traditional two-dimensional drawings with a symbolic meaning not machine-interpretable. The predominant open standard to exchange such BIM models is the Industry Foundation Classes (IFC) [33]. IFC follows a schema defined in the EXPRESS modeling language [24]. The IFC standard includes two text-based serialization formats to exchange instance models. These are based on parts 21 and 28 of the EXPRESS standard and describe a succinct ASCII exchange structure (IFC-SPF) and an XML-based structure respectively. These exchange formats do not impose an ordering or structure on the way instances are laid out in the file and do not support random access operations [26] and are, therefore, not suitable as storage systems for a high-performance query engine.

Special-purpose querying languages for the explicit semantics encoded in such building models have been proposed [34,48,13]. These are either intended for answering specific queries or rely on novel query syntaxes whose functionality is not as complete and their semantics not as precisely defined as query languages that went through extensive development and standardization processes, such as SPARQL (SPARQL Protocol and RDF Query Language) and SQL (Structured Query Language, the standard for relational databases).

With the introduction of the *ifcOWL*[1] ontology for building models [41], a reinterpretation from native IFC models to RDF (Resource Description Framework, a general information modeling method) is available and, subsequently, building models can be queried using SPARQL. SPARQL is a well-established language with precisely understood semantics [43] and capable of matching arbitrary graphs and returning attributes and transformations thereof. Examples of SPARQL

* Corresponding author.
*E-mail addresses:* t.f.krijnen@tue.nl (T. Krijnen), beetz@caad.arch.rwth-aachen.de (J. Beetz).
[1] http://www.buildingsmart-tech.org/future/linked-data/linked-data.

https://doi.org/10.1016/j.autcon.2018.07.014
Received 24 November 2017; Received in revised form 20 June 2018; Accepted 16 July 2018

```
#11652=IFCSITE('27TOPmxCrDgPimmYFfwtvD',#1,'Default',$,'',#9570,$,$,
↪  .ELEMENT.,(42,21,30,344238),(-71,-3,-35,-194702),-0.0,$,$);
```

**Listing 1.** `IfcSite` definition from the *office* model in prevalent text-based IFC-SPF encoding. Attributes are specified in a predefined order starting with a global identifier, ending with elevation. Omitted fields are indicated with "`$`", instance references are established with numeric identifiers and "`#`".

queries in a context relevant to the construction industry are provided in Appendix A on page 24. These form the basis of the performance evaluation throughout this paper.

The *ifcOWL* ontology is directly derived from the EXPRESS schema and favors compatibility with the original schema over an idiomatic OWL ontology (Ontology Web Language, a standard for authoring ontologies that builds upon RDF). As a consequence, the mapping from EXPRESS to OWL results sometimes in unidiomatic and verbose definitions, in particular for some of the geometry definitions [40].

The objectified relationship model and low-level nature of IFC renders many interesting relationships implicit, such as spatial connectivity and type information. Therefore, additional inference steps are often applied to increase its ease of use, querying efficiency and the discoverability of information, see for example de Farias et al. [17]and Pauwels et al. [39].

Listings 1 and 2 compare the verbosity of both SPF and RDF for a particular *IfcSite* definition. For the latter, in order to conserve space and eliminate redundancies, only two out of nine of the non-omitted attributes are provided. These two attributes are chosen to highlight the additional indirections introduced by the conversion. See Query 1 in the Appendix for a SPARQL query that operates on this type to find the *latitude associated to the IfcSite*.

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>    .
@prefix list:     <https://w3id.org/list#>                         .
@prefix ifcowl:   <http://ifcowl.openbimstandards.org/IFC2X3_TC1#> .
@prefix express:  <https://w3id.org/express#>                      .

inst:IfcSite_11652
 rdf:type ifcowl:IfcSite;
 ifcowl:globalId_IfcRoot inst:IfcGloballyUniqueId_95039 ;
 ifcowl:refLatitude_IfcSite inst:IfcCompoundPlaneAngleMeasure_95040 .

inst:IfcGloballyUniqueId_95039
 rdf:type ifcowl:IfcGloballyUniqueId;
 express:hasString "27TOPmxCrDgPimmYFfwtvD".

inst:IfcCompoundPlaneAngleMeasure_95040
 rdf:type ifcowl:IfcCompoundPlaneAngleMeasure ;
 list:hasContents inst:INTEGER_95044 ;
 list:hasNext inst:IfcCompoundPlaneAngleMeasure_95041 .

inst:IfcCompoundPlaneAngleMeasure_95041
 rdf:type ifcowl:IfcCompoundPlaneAngleMeasure ;
 list:hasContents inst:INTEGER_95045 ;
 list:hasNext inst:IfcCompoundPlaneAngleMeasure_95042 .

inst:IfcCompoundPlaneAngleMeasure_95042
 rdf:type ifcowl:IfcCompoundPlaneAngleMeasure ;
 list:hasContents inst:INTEGER_95046 ;
 list:hasNext inst:IfcCompoundPlaneAngleMeasure_95043 .

inst:IfcCompoundPlaneAngleMeasure_95043
 rdf:type ifcowl:IfcCompoundPlaneAngleMeasure ;
 list:hasContents inst:INTEGER_95047 .

inst:INTEGER_95044 rdf:type express:INTEGER ;
 express:hasInteger 42 .

inst:INTEGER_95045 rdf:type express:INTEGER ;
 express:hasInteger 21 .

inst:INTEGER_95046 rdf:type express:INTEGER ;
 express:hasInteger 30 .

inst:INTEGER_95047 rdf:type express:INTEGER ;
 express:hasInteger 344238 .
```

**Listing 2.** Only two attributes of the same `IfcSite` definition serialized in Terse RDF Triple Language (Turtle) according to Ref. [41]. In this format, attributes are not qualified by their order, but by a URI (Uniform Resource Identifier) such as `ifcowl:globalId_IfcRoot`. The **inst** prefix denotes a namespace specific to the model. https://github.com/IDLabResearch/IFC-to-RDF-converter.