Automation in Construction 92 (2018) 214–229

Contents lists available at ScienceDirect

Automation in Construction

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

# A rule-based methodology to extract building model views

Tarcisio Mendes de Farias[a,b,c,*], Ana Roxin[c], Christophe Nicolle[c]

[a] Department of Computational Biology, University of Lausanne, Lausanne, Switzerland
[b] SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland
[c] Laboratory LE2I, CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté (UBFC), Dijon, France

ARTICLE INFO

*Keywords:*
ifcOWL
Rule checking
Model view definition (MVD)
Reasoning
Semantic web

ABSTRACT

In this paper, we present a novel approach called *IfcView* that relies on Semantic Web technologies for creating building views. To do so, we consider an ifcOWL ontology proposed by buildingSMART. The ifcOWL is an Industry Foundation Classes (IFC) based ontology. By combining the ifcOWL ontology with logical rules (expressed in Semantic Web Rule Language, SWRL), we demonstrate through several case studies that our approach can perform a more intuitive and flexible extraction of building views when compared to the Model View Definition (MVD) approach. This is because our rule-based approach dynamically creates sub-graphs (i.e. views) by specifying the IFC elements to extract as Globally Unique Identifiers (GUID), relationships or entities. Another benefit of our approach is the fact that it simplifies the maintenance and definition of building views. Once our rule-based system extracts such a building view (i.e. sub-graph), this view can be exported by using STEP (STandard for the Exchange of Product) or Turtle (a Resource Description Framework (RDF) syntax) formats.

## 1. Introduction

A building life-cycle mainly comprises the following stages: the initial planning and design, the construction of the facility, the operation and maintenance and the dismantling and recycling of buildings and engineering structures. The data produced throughout the building's life-cycle is handled and updated by several different actors each intervening in different associated processes. BIM (Building Information Modeling) [1] is one of the latest approaches proposed in the AEC (Architecture, Engineering and Construction) domain for bridging the existing interoperability gap among systems in this field. A BIM system is a tool that allows users to integrate and reuse building-related data along with pertaining domain knowledge, throughout the considered building's life-cycle [2].

The first step in BIM standardization was conducted in 1999 by buildingSMART International (bSI) organization (formerly International Alliance for Interoperability, IAI) [3]. It resulted in the development of a model for representing all components of a physical building, namely the IFC model (Industry Foundation Classes). Unlike previous formats such as DXF (Drawing eXchange Format) [4] or DWG (DraWinG) [5], which were graph- and respectively vector-oriented representation formats, the IFC standard (ISO 10303-21) relies on object-oriented modeling (i.e. EXPRESS language based) [6].

In the context of BIM, a digital representation of the building comes in the form of one or several IFC files, therefore ensuring interoperability among the data produced with the various software tools used by the different actors from the AEC domain. Still, manipulating the data contained inside such IFC-based building representations remains a fastidious process, mainly performed manually (by selecting IFC elements) and therefore source of numerous errors. Notably, in order to facilitate the handling of such IFC files, there is an increasing need to display only the information pertaining to a given business logic or context (i.e. view). This need also arises when a given stakeholder wishes to update or to modify the information contained in an IFC file, and then forward the result to another stakeholder, or insert it into a global BIM representation. Such workflow can be compared to loosely coupled federated architectures as defined in Ref. [7]. Thus, in our vision, BIM stands as a cooperative system of unified views of the same building. Each stakeholder is allowed to locally keep a view of the global building model. Such view is defined as the sum of necessary and sufficient building information needed for the correct realization of their related business processes (e.g. plumbing, building renovation, window cleaning). In other words, to deliver such a view, one has to correctly extract the minimal sub-graph of elements from the IFC file(s) representing the whole building.

Given the fact that organizations (such as bSI, but also the International Organization for Standardization, ISO) have been recognized Semantic Web technologies as a potential mean to solve issues related to the handling of IFC files, the approach we present here is solely based on the OWL version of the IFC standard (i.e. ifcOWL

* Corresponding author.
*E-mail addresses:* tarcisio.mendesdefarias@unil.ch (T.M.d. Farias), ana-maria.roxin@u-bourgogne.fr (A. Roxin), cnicolle@u-bourgogne.fr (C. Nicolle).

https://doi.org/10.1016/j.autcon.2018.03.035
Received 27 July 2017; Received in revised form 2 March 2018; Accepted 29 March 2018

ontology, as pushed by bSI [8]). Our goal is to demonstrate that Semantic Web technologies are suited for extracting pieces of data from building models. We aim at proving that this approach allows a more intuitive extraction of building views and mitigates the gap of semantic heterogeneity for building software interoperability. Indeed, our approach is a rule-based system, where rules are expressed by means of terms either extracted from the ifcOWL ontology or inferred (see Section 4 and Appendix A). In other words, our main focus is to show that Semantic Web technologies can be used to efficiently (semi)-automatically extract building views.

Section 2 begins with a brief overview of the most pertaining related works in the context of the ifcOWL ontology and building model views. Section 2 also includes a review of several recent rule-based systems as implemented in a BIM context. Section 3 depicts Semantic Web concepts that are needed to understand the rest of the work presented here. Section 4 briefly describes how we create the knowledge graph based on the ifcOWL ontology and how we define our logical rules to ease the model view extraction. Section 5 explains the implementation of our rule-based system prototype for extracting building views. Section 6 evaluates the implemented prototype in terms of query execution time and number of expected results. Finally, we conclude this article by discussing the benefits of such a rule-based system compared to traditional approaches. We end this article by enumerating several improvements that could be implemented in order to enhance the performance of our prototype.

## 2. Related work

This section covers three research fields. The first one is dedicated to the conversion of the IFC standard into an OWL ontology. The second one concerns the most relevant approaches for defining building model views. Finally, the third one presents two recent BIM systems that use logical rules.

### 2.1. IFC-to-OWL converters

IFC-to-OWL converters studied in this section rely on automatic methods for conceiving an OWL ontology from the EXPRESS schema of the IFC standard. Developed in the context of the IntelliGrid EU FP6 project [9], the approach described by Beetz et al. in [10] is the first well-defined approach to translate the IFC standard into OWL (i.e. ifcOWL). Over the last few years, various IFC-to-OWL converters were proposed (see related approaches in Refs. [10-15] for further details).

Dibley et al. in Ref. [16] present how to conceive an IFC ontology by parsing STEP[1] files that contain IFC entities. Unfortunately, this translation of the IFC standard into OWL only covers a limited set of IFC classes, namely the classes considered relevant for the project at hand (that is the implementation of an augmented environment for intelligent agents). Today, the reference (candidate standard) version of an ifcOWL ontology is proposed by buildingSMART and available in Ref. [8]. The IFC-to-OWL converter to generate the buildingSMART's ifcOWL version is fully described by Pauwels et al. in Ref. [11]. As a reminder, an IFC-to-OWL converter is a method for generating an OWL ontology starting from the IFC EXPRESS schema.

### 2.2. Defining model views

The approach defined in the context of the buildingSMART community for specifying sub-parts of IFC files is called Model View Definition (MVD). An MVD "*defines a subset of the IFC schema that is needed to satisfy one or many Exchange Requirements of the AEC industry*" [17]. The specification of MVDs is currently enabled through the IfcDoc[2] tool. At the moment, the MVD specification relies on modular Concept Templates [18], namely bricks of IFC schema snippets. After composing an MVD with the IfcDoc tool, the MVD can be exported as an mvdXML file [17], both for future reuse and for inclusion in an IDM (Information Delivery Manual). In addition, any IFC file can be validated according to an MVD loaded in the IfcDoc tool. In other words, the IfcDoc tool allows validating a given IFC file with respect to a given MVD. Thus the main issue with IFC files remains the information extraction. This is why, in this paper, we investigate to what extent Semantic Web technologies can be applied for extracting data from an IFC file. Our present scope does not cover using Semantic Web technologies for validating an IFC file. With this being mentioned, the MVD approach, as defined by bSI, has several main drawbacks listed below:

1. MVDs lack logical formalisms (e.g. description logic, horn-like rules and shape expressions or constraints) which can take advantage of reasoners and rule engines. For instance, such formalism eases the implementation of validation rules [19].
2. The MVD solely considers the IFC schema. Our approach (described in Section 4) goes beyond this, by addressing both IFC schema and instances' GUIDs (Globally Unique Identifier[3]), thus making building views' definition process more flexible.
3. An MVD-based constructor is based on static and previously defined XML files (mvdXML). These mvdXML files have to be entirely parsed before extracting a building view from an IFC project file.
4. In addition, in the MVD approach, a simple view modification implies reconstructing the entire mvdXML file.

Our approach relies on logical rules to express the elements that are part of a building view. These rules are dynamically applied during a view extraction by a rule engine. Thus, as rules can be easily added or removed from the knowledge base (see related definition in Section 3), our approach has a higher level of flexibility. It can be seen as a way to filter an IFC file according to a specific need or context. With our approach, all irrelevant data can be removed from the BIM before generating the ifcOWL file. The queries further addressed on this file are solely handled by Semantic Web technologies, thus allowing the two following advantages:

1. High maintainability: only one ifcOWL file derived from one master IFC model along with the set of queries or rules (as adapted from the considered mvdXML files) need to be maintained. All subset generation is based on ifcOWL graphs and will be done automatically on query time.
2. High modularity: one is able to intuitively generate new MVDs by combining queries and rules. Our approach thus matches the original MVD modularity; in our approach Concept Templates are represented as formal logical rules or queries.

In Ref. [20], the authors describe a partial building view extraction method (as RDF graphs and schemas), mostly based on a preliminary version of the ifcOWL ontology [21]. However, this method has two major limitations when compared with our approach. Firstly, this approach doesn't allow for generating a view as an IFC-STEP file. Thus, in other words, the output in Beetz et al.'s approach cannot be used as an input to Computer-aided design (CAD) software tools handling IFC files. Secondly, the authors in Ref. [20] do not propose a semi-automatic way to define building model views. Actually, the building view is defined by manually writing a unique SPARQL CONSTRUCT[4] query.

[1] https://www.iso.org/standard/63141.html.

[2] http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool.

[3] http://www.buildingsmart-tech.org/implementation/get-started/ifc-guid.

[4] https://www.w3.org/TR/rdf-sparql-query/#construct.