A Rule Based System for Semantical Enrichment of Building Information Exchange

Farias[1], M. T., Roxin[2], A. & Nicolle[2], C.

[1]Active3D, Dijon, France
t.mendesdefarias@active3D.net
[2]Checksem, Laboratory LE2I (UMR CNRS 6306), University of Burgundy, Dijon, France
{ana-maria.roxin,cnicolle}@u-bourgogne.fr

Abstract. In the area of building construction and management, the dematerialization of data and processes has been a global issue for the past 10 years. Going beyond the geometric representation of a building, Building Information Modeling (BIM) is an approach that aims at integrating into one single system heterogeneous data and processes from different actors. Such integration is a complex and fastidious task. The implementation of the related processes for data querying, retrieval or modification is not less difficult. To tackle this problem, we have developed a novel approach based on Semantic Web technologies. In doing so, we have defined an ontology inspired on IFC standard for representing building information. On top of this ontology, we have defined and implemented a set of SWRL rules. The paper at hand describes these rules and their application in the context of building information handling (notably by means of IFC files).

Keywords: BIM, IFC, semantic web, ontologies, SWRL, SPARQL, OWL, AEC.

1 Introduction

A building lifecycle mainly comprises two phases: building construction and facility management. The data produced throughout the building's lifecycle is handled and updated by several actors intervening in the associated processes. BIM (Building Information Modelling) [1] is one of the latest approaches proposed in the AEC (Architecture, engineering and construction) domain for bridging the existing interoperability gap among systems existing in this field. In the context of our approach, we define BIM as the process of generating, storing, managing, exchanging and sharing building information in an interoperable and reusable way. 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 lifecycle [2].

The first step in BIM standardization was conducted in 1999 by buildingSMART (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) [1, 6] relies on object-oriented modeling.

In the context of BIM, a digital representation of the building comes in the form of one or several IFC files, therefore ensuring the interoperability among data produced with the various software tools used by the different actors from the AEC domain. Still, manipulating the data contained inside IFC-based building representations remains a fastidious process, mainly performed manually 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. 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 [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 (etc. plumbing, building renovation, window cleaning). In other words, for delivering such a view, one has to correctly extract the minimal subgraph of elements from the IFC file(s) representing the whole building.

In order to address this issue, we propose a novel approach based on Semantic Web technologies where the IFC standard is represented as an OWL ontology. It allows a more intuitive extraction of building views and mitigates the gap of semantic heterogeneity for building software interoperability. This paper begins with a brief overview of Semantic Web technologies. Then, we present the related works in the section 3. After a brief description of our IFC ontology, we present three main benefits for applying SWRL rules in the context of BIM. We conclude this article by arguing the benefits of SWRL rules to tackle above-mentioned BIM-related issues.

2 Semantic Web Technologies

One of the main visions of the future of the Web is the vision of the Semantic Web or Web of Data. This vision allows machines to understand the data they manipulate (by means of standard data descriptions) and gives a formal definition of Web resources (by means of ontologies). These two mechanisms make it possible to automatically reason over large scale Web repositories and infer new information from existing data sources (by means of reasoners).

Languages developed for the Semantic Web are based on metadata models (RDF/RDF Schema [8]) and logic-based knowledge representation (the Web Ontology Language [8]).

Ontology languages are based on Description Logics (DL) formalisms, meaning that a concept is defined through an ensemble of necessary and sufficient conditions. Therefore, an ontology comprises a terminological model (`TBox`) which contains the formal definitions of the concepts relevant for the considered domain of discourse. The instantiation of these concepts results in an assertional model (ABox). The combination of terminological and assertional boxes results in a so-called knowledge base.

In a rule-based system, rules are expressed by means of terms defined in ontologies. Rule languages are a complement to DL-based languages (such as OWL) as they allow representing different axioms. The development of such rule languages has started in 2000, with the RuleML initiative [9]. Based on the Logic Programming paradigm, RuleML implements a RDF syntax. The Semantic Web Rule Language (SWRL) [8, 10] is based on Logic Programming as well, but combines OWL and RuleML. SWRL makes it possible to specify conjunctive rules over the concepts and relationships present in an OWL ontology. Generally, ontologies address taxonomic reasoning problems (data classification), whereas rule-based systems aim at reasoning problems involving data [8].

3 Related Works

Developed in the context of the IntelliGrid EU FP6 project [11], the approach described by Beetz et al. in [12] is one of the mostly used approaches for translating the IFC standard into OWL (IfcOWL). The authors present an automatic method for conceiving such OWL ontology from the EXPRESS schema of the IFC standard. However, all IFC "Defined Types" (more than one hundred) are mapped as OWL classes, which significantly increases the number of triples when publishing the so-generated knowledge base (OWL model and concept instances) into a triple store (a system for storing and querying RDF databases [13]).

Dibley et al. [14] present how to conceive an IFC ontology by parsing STEP files that contain the IFC schemas. 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).

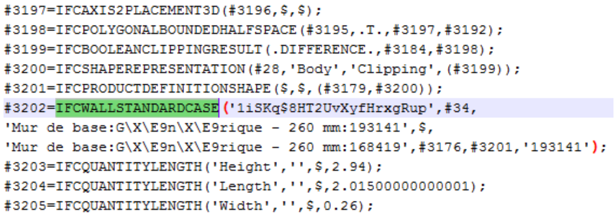
Pauwels et al. in [15] propose the automated checking of building environment regulations (e.g.: the acoustic performance regulations) using N3 Logic [16] rules applied to IfcOWL[12] ontology. In the paper [17], the authors deal with the interoperability of 3D information from IFC to the standards: X3D (Extensible 3D Graphics)[17] and SLT (STereoLithography)[17]. The latter approach represents these standards as ontologies (e.g.: IfcOWL) and makes available geometric data from IfcOWL to the SLT and X3D ontologies by means of N3 Logic rules. However, to the best of our knowledge in the literature, there are not works which point out the enrichment of the IFC model without compromise the system interoperability and allowing the coexistence of data from different versions of IFC files dynamically.

The next section gives a few considerations regarding the conceived ontology inspired on IfcOWL[12] where we modify the translation of "Defined Types" and collections to conceive a more suitable ontology. After, we mostly illustrate the key benefits obtained by implementing SWRL rules on top of this ontological model of the IFC standard.

4 Benefits of applying SWRL rules to handle IFC files

The creation of a knowledge base for manipulating building-related information implies several steps:

1. The first step is the conception of the ontological model from the IFC standard model. For doing so, we have developed an automatic parser that builds the terminological box (`TBox`) of the knowledge base starting from the EXPRESS specification of the IFC standard (downloaded from [6]). The translation rules used for generating this IFC ontology are inspired on IfcOWL[12] and the Table 1 shows the main modifications made in this approach;
2. Once this model has been created, we use software tools (such as Protégé [18]) which provide a simple interface for the ontology engineer to define SWRL rules on top of the `TBox.` The rules are created following the expressivity needed by our clients and audited by our AEC/FM (Architecture, Engineering and Construction/Facility Management) staff;
3. The resulting terminological model is then uploaded onto a triple store. In the context of our approach, we have used the Stardog RDF database because it supports user-defined rule reasoning (SWRL) and has mechanisms to deal with rule inconsistency [19];
4. For populating the so-built `Tbox` with information from IFC files (see an example in the figure 1), we have developed a parser that extracts data from an IFC file and creates the respective concept instances in the knowledge base. In the IFC file syntax, the list of parameters between parentheses for each entity is the correspondent values for the OWL class properties. During this process, we use one repository per IFC file.



```
#3197=IFCAXIS2PLACEMENT3D(#3196,$,$);
#3198=IFCPOLYGONALBOUNDEDHALFSPACE(#3195,.T.,#3197,#3192);
#3199=IFCBOOLEANCLIPPINGRESULT(.DIFFERENCE.,#3184,#3198);
#3200=IFCSHAPEREPRESENTATION(#28,'Body','Clipping',(#3199));
#3201=IFCPRODUCTDEFINITIONSHAPE($,$,(#3179,#3200));
#3202=IFCWALLSTANDARDCASE('1iSKq$8HT2UvXyfHrxgRup',#34,
'Mur de base:G\X\E9n\X\E9rique - 260 mm:193141',$,
'Mur de base:G\X\E9n\X\E9rique - 260 mm:168419',#3176,#3201,'193141');
#3203=IFCQUANTITYLENGTH('Height','',$,2.94);
#3204=IFCQUANTITYLENGTH('Length','',$,2.01500000000001);
#3205=IFCQUANTITYLENGTH('Width','',$,0.26);
```

Fig. 1. A portion of an IFC file where an instance of IFCWALL entity is defined.

The following sections explain the benefits achieved by defining and using SWRL rules for the processing of such IFC-extracted data.

Table 1. Modified rules for the translation in OWL language of the IFC standard.

Mapping rule description	**Example**
Collections	

If the order is not important, a collection (LIST) is mapped as values of non-functional OWL property else different properties are created.	The IFC attribute “coordinates” is an ordered list of three elements. Then, it is mapped as three OWL properties: coordinateX, coordinateY and coordinateZ.
IFC Attributes	
We map the IFC attributes to an OWL object-property.	The attribute OwnerHistory is mapped as an OWL object and functional property.
Defined Types	
The Defined Types which compose an IFC relation are not mapped directly as classes. In the ontology, we merge these Defined Types that have the same data type in one class. A property describes the semantic of the value in the IFC context (e.g.: the property ifc:name has an annotation property hasIfcType with the string value “IfcIdentifier”).	IfcVolumeMeasure, IfcAreaMeasure, IfcLengthMeasure and others are mapped as a class named Real because they are indeed real values. The class Real has the property ifc:hasRealValue with a range xsd:double. In the case of the IfcSimpleProperty class, we define a property to describe the semantic of its value in the IFC context (e. g: IfcVolume.).
Some Defined Types are indeed enumerations described in the WHERE clause within the SELF IN reserved word. The SELF IN enumeration is mapped as the range of a new property.	IfcTextAlignment has the WHERE clause : SELF IN ['left', 'right', 'center', 'justify']. Then, IfcTextAlignment is mapped as the data property hasTextAlignment with the range {“left”, “right”, “center”, “justify”}.

4.1 Dynamic classification of eligible IFC components in a specific industrial context

In many application contexts, the IFC standard fails to deliver a proper solution tailored to the exact needs of the building actors. For example, the facility manager will not directly handle IFC entities, but will search for façade walls, determine spaces that cannot be accessed, or ask the system for the number of rooms in a building. This information is implicitly described in the IFC model. However, it is not easily exploitable when relying exclusively on the IFC standard, because this implies complex calculus that is based solely on the geometry of the elements contained in the IFC file.

In order to address these limitations, our approach allows defining novel concepts as used by AEC/FM actors by means of SWRL rules. A set of rules allow specifying additional information regarding the concepts defined in the **TBox**. Notably, they specify means for discovering and generating new knowledge from the exiting one. Therefore, the implicit information becomes explicit.

For example, consider the case of a facility manager that needs to plan the cleaning of all windows of a given building. The concept of a windowed-space is not present in the IFC standard, so identifying such spaces would represent a lot of manual work from the facility manager. However, this information can be easily exploited, if we create the concept **BimSpaceWithWindow** populated by the following SWRL rule which identifies spaces that have one or more windows:

```
IfcRelSpaceBoundary(?x) & IfcSpace(?y) & IfcWindow(?z) & RelBuildingElement(?x, ?z) & RelSpace(?x, ?y) ⇒ BimSpaceWithWindow(?y).
```

We can easily extend this example to the case where SWRL rules are used to specify precise business contexts and processes.

4.2 Simplifying the writing of SPARQL queries

Defining SWRL rules on top of our `TBox` gives the advantage of simplifying the writing of SPARQL queries because it offers a more fine-grained ontology. For example, let us consider the following SPARQL query that allows retrieving all external walls of a building:

```
SELECT ?externalWall WHERE {
?externalWall a ifc:IfcWall.
?o a ifc:IfcDefinesByProperties;
ifc:RelObjects ?externalWall;
ifc:RelPropertyDefinition ?pSet.
?pSet a ifc:IfcPropertySet;
ifc:HasProperties ?p.
?p a ifc:IfcPropertySingleValue;
ifc:Name  ?name.
?name ifc:dp_IfcIdentifier
"IsExternal".
?p ifc:NominalValue ?val.
?val a ifc:IfcBoolean;
ifc:dp_IfcBoolean
"true"^^xsd:boolean}.
```

After considering the definition of a rule on top of our ontological model that specifies, using existing ontology concepts, what is an external wall. Such a rule can be defined as follows:

```
ifc:HasProperties(?a, ?x) & ifc:NominalValue(?x, ?z) & ifc:Name(?x, ?y)
& ifc:RelPropertyDefinition(?b, ?a) & ifc:RelObjects(?b, ?c) &
ifc:IfcWall(?c) & ifc:dp_IfcBoolean(?z, true) & ifc:dp_IfcIdentifier(?y,
"IsExternal") ⇒ BimExternalWall(?c).
```

This rule populates the concept of `BimExternalWall` which does not exist in the IFC standard. By integrating the concept of `BimExternalWall` defined through the SWRL rule, we greatly simplify the writing of the SPARQL query, making it more comprehensible and understandable:

```
SELECT ?externalWall WHERE { ?externalWall a ifc:BimExternalWall.}.
```

4.3 Dynamic handling of the IFC standard's evolution

Starting with the publication of the first version of the IFC standard [3, 6], this specification has gone several updates and modifications. Generally, there is no backward support between the different versions, as illustrated by the IFC change log [6].This is mainly due to the fact that most modifications are made in the data model structure: modifying the attributes' order for a given IFC entity, replacing a deleted entity with another data structure, etc.

As an example, let us consider the entity IfcAnnotationSymbolOccurrence that was deleted in the IFC2x4 RC2 version [6]. Still, its supertype (superclass) IfcStyledItem has been preserved. If we consider the case in which an IFC2x4 file has to be published on the triple store, we have to ensure that it respects our ontological model.

Therefore, we have to define a SWRL rule for handling IfcStyledItem instances from the IFC2x4 file that belonged to the IfcAnnotationSymbolOccurrence concept in the IFC2x3 version [6]. For doing so, we define the following SWRL rule:

```
ifc:IfcStyledItem(?x) & ifc:Item(?x, ?y) & ifc:IfcDefinedSymbol(?y)⇒
IfcAnnotationSymbolOccurrence(?x).
```

Another example of application is the case when standard evolution adds new entities as subclasses of existing IFC entities. For example, the IFC2x4 version has added the entity IfcPipeSegment as a subclass of the entity IfcFlowSegment. Previous versions of the IFC standard do not explicitly mention the "pipe segment" concept, but they contain all the necessary information pertaining to this new concept. We may therefore define an SWRL rule that will be applied over instances already contained by our knowledge base:

```
IfcFlowSegment(?a) & IfcDefinesByType(?b) & RelatedObjects(?b,?a) &
RelatingType(?b,?c) & IfcPipeSegmentType(?c) ⇒ IfcPipeSegment(?a).
```

Therefore the triple store existing data is automatically restructured according to those new rules. Data extracted from IFC files complying to previous versions of the IFC standard can be automatically update in order to comply to the newer versions of the standard. These mechanisms allow us to handle different IFC schemas, thus increasing the interoperability of information exchange among stakeholders.

5 Conclusion and Future Work

The above described vision of a BIM-based information system aims at easing the processes and the information exchanges related to facility management and building construction. In order to achieve this vision, we have developed an ontological model for the IFC standard. In the context of BIM, the stakeholders use various software tools for different objectives and the interoperability is made by standards like IFC. Nevertheless, the needs of defining new concepts in the data model for a specific stakeholder context compromise the interoperability among those systems. To solve this problem, we have used web semantic technologies that allow us defining new concepts by means of rules separating the BIM data structure model (e.g.: IFC) from its semantic. Therefore, we can increase the data model expressivity without compromising the interoperability made by IFC files. The paper at hand has illustrated the main benefits of having developed such business-specific rules.

Further works will address the comparison of SWRL rules syntax to the one of Rule Interchange Format (RIF) as defined by the W3C [8]. Last but not least, the need for checking for inconsistencies or ambiguity by introducing new rules will be also considered besides the mechanism offered by Stardog to deal with it.

This work is part of a collaborative project with the French company ACTIVe3D[1] who has financed this work.

[1] http://www.active3d.net/fr/

6 References

1. Volk R., Stengel J., Schultmann F.: Building Information Modeling (BIM) for existing buildings - Literature review and future needs. Automation in Construction, Elsevier journal, Volume 38, pp 109-127 (2014)
2. Lee G., Sacks R. & Eastman Charles M.: Specifying parametric building object behavior (BOB) for a building information modeling system, Automation in Construction, Elsevier, Volume 15, Issue 6, November, Pages 758-776, Knowledge Enabled Information System Applications in Construction, (2006)
3. Vanlande, R., Nicolle, C., & Cruz, C.: IFC and building lifecycle management. Automation in Construction, Elsevier journal Volume 18(1), pp. 70-78 (2008)
4. Autodesk, Inc.: DXF Reference. San Rafael, USA: Autodesk, Inc (2011)
5. Open Design Alliance: Open Design Specification for .dwg files (2013), `http://opendesign.com/files/guestdownloads/OpenDesign_Specification_for_.dwg_files.pdf`
6. International Alliance for Interoperability: IFC2x Versions (2013), http://www.buildingsmart-tech.org/specifications/ifc-overview
7. Sheth A. P. & Larson J. A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys, Volume 22, N° 3 (1990)
8. The World Wide Web Consortium (W3C), `http://www.w3.org/`
9. RuleML, `http://www.ruleml.org`
10. Horrocks, I., Patel-Schneider, P. F., Bechhofer, S. & Tsarkov, D. (2005). OWL rules: A proposal and prototype implementation. Journal of Web Semantics, 3(1):23-40
11. Gehre, A., Katranuschkov, P., Wix, J., & Beetz, J.: Interoperability of Virtual Organizations on a Complex Semantic Grid. Dresden: InteliGrid (2006)
12. Beetz, J., Leeuwen, J. V., & Vries, B.: IfcOWL: A case of transforming EXPRESS schemas into ontologies. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), 89-101 (2009)
13. Allemang, D. & Hendler, J.: Semantic Web for the Working Ontologist. (M. Kaufmann, Ed.) San Francisco: In Dean Allemang and James Hendler (2008)
14. Dibley, M. J.: An Intelligent System for Facility Management. Wales, UK: Cardiff School of Engineering, Cardiff University (2011)
15. Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R. & Van Campenhout, J.: A semantic rule checking environment for building performance checking. Automation in Construction 20(5), 506-518 (2011).
16. Tim Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y. & Hendler, J. N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming, 8, pp 249-269 (2008).
17. Pauwels, P., Van Deursen, D., De Roo, J., Van Ackere, T., De Meyer, R., Van de Walle, R. & Van Campenhout, J.: Three-dimensional information exchange over the semantic web for the domain of architecture, engineering, and construction. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 25(4) 317-332 (2011).
18. Protégé, `http://protege.stanford.edu/`
19. Stardog version 2.1.3, `http://stardog.com/`