SimpleBIM: From full ifcOWL graphs to simplified building graphs

P. Pauwels
Dept. of Architecture and Urban Planning, Ghent University, Ghent, Belgium

A. Roxin
LE2I Laboratory (UMR CNRS 6306), University of Burgundy, Dijon, France

ABSTRACT: Recent research in semantic web technologies for the built environment has resulted in several proposals to further improve information exchange among stakeholders from the domain. Most notable is the production of several OWL ontologies that allow to capture building data in RDF graphs. For example, an ifcOWL ontology allows to capture IFC data in an RDF graph. As the building data is now available in a semantic graph with an explicit formal basis, it can be restructured and simplified so that it more easily matches the different requirements associated with practical use case scenarios. In this paper, we investigate several proposals and technological approaches to simplify ifcOWL building data, thus addressing the needs of specific industrial use cases.

1 INTRODUCTION

Semantic web technologies have been identified as promising in the field of AEC (Architecture, Engineering, Construction), notably for leveraging the different challenges related to BIM (Building Information Modeling) (Mendes et al. 2015a). Recent research in this domain has resulted in several proposals to further improve information exchange among stakeholders from the domain. Among those efforts, most notable are the production of ifcOWL (Pauwels & Terkaj 2016) and ifcWoD (Mendes de Farias et al. 2015b) ontologies. These ontologies allow the publication of IFC-based building models as directed labelled graphs, represented using the Resource Description Framework (RDF) data model (buildingSMART International 2015a). The resulting RDF graphs can be published as part of a linked building data cloud and can be processed by the latest semantic web technologies, which includes open publication platforms (triple stores), query services (SPARQL endpoints), and inference engines for reasoning.

At the moment, however, the above mentioned ontologies remain close to the original IFC schema as available in the EXPRESS information modelling language. This is for example one of the key criteria outlined in Pauwels & Terkaj (2016) and in Mendes et al. (2014). This criterion is maintained in order to keep the EXPRESS, XSD and OWL schemas as identical as possible, with the IFC EXPRESS schema forming the master schema. This has two key consequences:

1. Many of the EXPRESS-specific semantic constructs (like SELECT data types, LIST data types) are maintained and result in complex and unintuitive constructs in OWL and RDF.
2. The instance graphs (ABox) are at least as large and complex as the original IFC models.

Thus, the current ifcOWL ontology does not really simplify handling IFC models as it does not deliver the highly demanded simpler models to AEC practitioners. In other words, the proposed ifcOWL ontology is necessary as an OWL ontology that is nearly identical to the master EXPRESS schema, but in practical engineering use cases, simpler RDF graphs are desirable.

In this article, we therefore look into the generation of simpleBIM models (as RDF graphs) starting from the ifcOWL ontology. Such simpleBIM models in RDF should be more agile than what is available today. In our paper, we first provide an overview of related work (Section 2), which includes a brief overview of the Model View Definitions (MVD) approach and an overview of related suggestions in the Semantic Web domain. Suggestions in the Semantic Web domain typically include simple ontologies to represent buildings without the usage of IFC. We then describe a simple 3D building model (Section 3), for which an RDF graph is generated according to the ifcOWL ontology. Several simplifications of the RDF graph are proposed, resulting in clearly defined approaches to simplify the BIM model. In Section 4, we briefly outline the possible approaches in implementing the transition towards simpleBIM models, which includes (1) straightforward programming using the available semantic web technology APIs (Jena, dotNetRDF), (2) the usage of SPARQL queries, and (3) the usage of logical rules and an OWL-DL based inference engine. We focus mainly on the resulting simplifications that can be made, however. In the concluding section, an overview is given of how the proposed approach can be combined with existing tools and information handling approaches, after which future steps are proposed.

2 RELATED WORK

2.1 Model view definitions (MVDs)

Some proposals have already been made in terms of simplifying building information in an IFC syntax. Within the AEC industry, the most closely related proposal in this regard is the Model View Definition (MVD) approach. This approach is tightly related to the usage of Information Delivery Manuals (IDMs). An IDM aims to methodologically *“capture and specify processes and information flow during the lifecycle of a facility”* (Karlshøj 2011). As such, an IDM supports and assures the efficiency and effectiveness of the AEC business processes described in the IDM. By the combination of an IDM with formally specified MVDs, a structured and well-controlled information exchange can take place among AEC project stakeholders, thus greatly improving the internal cost model of AEC projects.

An MVD in itself formally describes a subset of the full IFC schema, including additional data requirements. This subset is meant to respond to one or more specific Exchange Requirements (ERs) in an Information Delivery Manual (IDM). MVDs can be formally represented using mvdXML files. Not only does an mvdXML formally specify what should be contained in an MVD (*specification*), it can also be used to *validate* to what extent an IFC file actually contains the information specified in an MVD.

The specification of MVDs is currently enabled through the IfcDoc tool (buildingSMART International 2015b). At the moment, the MVD specification relies heavily on modular Concept Templates (Venugopal et al. 2012). These are blocks of IFC schema snippets (e.g. all address information for any IFC object), which can readily be selected when specifying a full MVD in the IfcDoc tool. After composing an MVD in the IfcDoc tool, it can be exported as an mvdXML file, both for future reuse and for inclusion in an IDM. In addition, any IFC file can be validated according to the loaded MVD, which means that the IfcDoc tool visually indicates to what extent the information specified in the MVD is also present in the IFC file.

It is important to point out that an mvdXML file is mainly used to *specify* and *validate* MVDs, and not necessarily to *generate* MVDs. In other words, the main purpose is that the MVD specifies which data needs to be exchanged, after which it is the responsibility of implementers and end users to also generate IFC models that comply with this MVD. So, it could be that the end user generates a full IFC model along with more specific and more limited MVD-compliant IFC models in the STEP Physical File Format (SPFF), but the latter is not necessarily a subset from the former, let alone that the latter is generated from the former. In summary, the mvdXML file is a formal representation schema, as well as an implementation guide as well as a validation tool.

Instead of limiting the role of mvdXML as an implementation guide only, one could also consider an mvdXML file as a *generation* tool, which generates a subset IFC model starting from the full IFC model. As an example, one could in this case implement a simple parser that loads an mvdXML file into an IFC file editor (or BIM authoring environment), after which a small subset IFC file is output that only includes the information as required according to the mvdXML specification.

This is precisely the approach that was proposed in Weise & Pauwels (2015), relying however on semantic web technologies. In Weise & Pauwels (2015), the mvdXML specifications of particular MVDs are translated into logical rules that are combined with an ifcOWL instance graph and ontology, and a semantic inference engine, so that subset IFC graphs are output that are compliant with the mvdXML specifications.

2.2 Simple BIM ontology

An entirely distinct strategy to simplify BIM information models, is to entirely disregard the IFC standard and instead consider drastically simplified BIM ontologies. Several authors have suggested such ontologies. For example, Niknam and Karshenas (2015) proposed a 'sumo' shared ontology that only contains the key components of a building model (walls, spaces, elements, floors). Depending on the use case, this shared ontology is then expanded with data following a design ontology, an estimating ontology, and so forth. A much earlier example in which a separate ontology was built from scratch, aiming particularly to represent building knowledge in an ontology-based fashion, was proposed by Lima et al. (2003, 2005) as part of the e-Cognos project. This ontology describes four key elements in construction, namely actors, resources, processes and products. Many of the lessons learnt from the e-COGNOS project are documented in El-Diraby (2013), which presents a domain ontology for construction knowledge (DOCK 1.0) starting from the earlier e-COGNOS work (2005-2013). Similarly, Ruikar et al. (2007) proposed an extensible set of modular ontologies (design-process ontology and team profile ontology) which are then deployed in an ontology-based knowledge-sharing environment (OnToShare) for usage by various stakeholders in construction industry.

Scherer & Schapke (2011) propose a multi-model driven construction management system with a layered ontology framework at the center. The ontology framework includes a Project Collaboration Ontology (PCO), which is composed of 5 sub-ontologies: a Construction Core Ontology (CCO), a multi-media visualization ontology, a software service ontology, an organization ontology, and an information process ontology. Dibley et al. (2012) propose an OntoFM system for real-time building monitoring, which relies on a building ontology based on IFC, a sensors ontology that relies on the OntoSensor ontology, and a general purpose ontology (SUMO) that captures domain independent concepts. Reinisch et al. (2011) and Kofler et al. (2012) developed a ThinkHome OWL ontology, including concepts related to resources (white and brown goods), building (layout, spaces, material), actors (schedules, preferences, context), energy (environmental impact, energy providers), comfort (thermal and visual), and exterior influences (weather, climate). The ThinkHome ontology and project relies heavily on the data coming from household appliances. This is inspired to some extent by the ontology-based household device models in the Domotic OSGi Gateway Ontology (DogOnt) (Bonino & Corno 2008). DogOnt allows only a simplified representation of a building as it focuses mainly on the home automation parts of its ontology.

Several similar proposals are made, their common element being the reliance on a simple building ontology: simpleBIM. In the remainder of this paper, we therefore investigate to what extent simplification strategies can be applied to ifcOWL models so that IFC content can better serve alternative use case that only require a simplified building representation.

3 SAMPLE CASE

For this paper, we have used a simple sample building model, which was modelled in Revit Architecture 2016 and which consists of a simple rectangular building that is located in a flat site. The building is furthermore divided in three spaces, has a floor and a roof and has several windows and floors, as displayed in Figure 1.

The building model was exported to IFC-SPF, in the IFC2X3_TC1 schema, which resulted in a 110kB file. The IFC-SPF was then converted into an RDF graph, using the IFC-to-RDF converter supplied in Pauwels, 2015. The file size of the resulting RDF graph is 767kB. The RDF graph counts 10,173 distinct triples (counted using Jena 3.0.0 and no inferences), and includes 5535 class instances (5580 after running the OWL inference engine). Note that 1313 instances are actually named individuals that are present in the ifcOWL ontology, representing ENUMERATION individuals, so the instance file actually contains only 4222 class instances. All files are made available for reference (Pauwels and Roxin 2016).

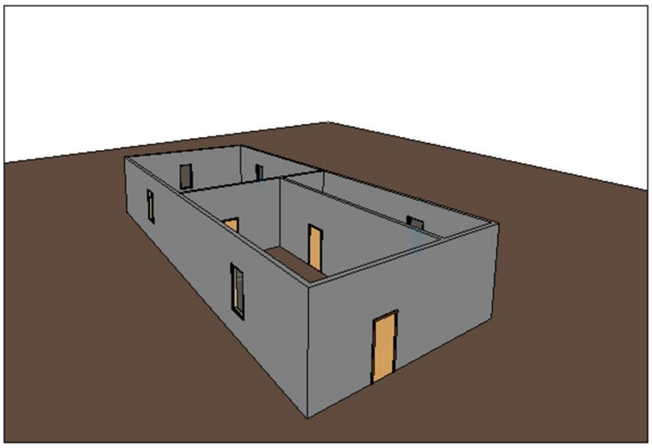

Figure 1. Display of the simple building model that was used here (visualized without roof).

The IFC building model in RDF follows the main structure of IFC. The core structure starts with a unique IfcProject instance, any valid IFC file allowing only one instance of this concept. This IfcProject instance can be considered as an aggregation point for the spatial structure of the building model. Figure 2 displays the RDF data structure that represents this data structure. As it can be noticed in Figure 2, a project comprises a number of sites (IfcSite); a site contains a number of buildings (IfcBuilding); a building has a number of building storeys (IfcBuildingStorey); finally a building storey comprises several spaces (IfcSpace). However, in the IFC standard, these relations are represented using intermediate IfcRelAggregates instances. The problem is that these instances are often unneeded in the eventual applications reusing or querying this information, thus their presence in the RDF graph raises its complexity unnecessarily. In the context of a graph approach, direct labelled relations between the different concepts (IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey, and IfcSpace) are preferred by far.

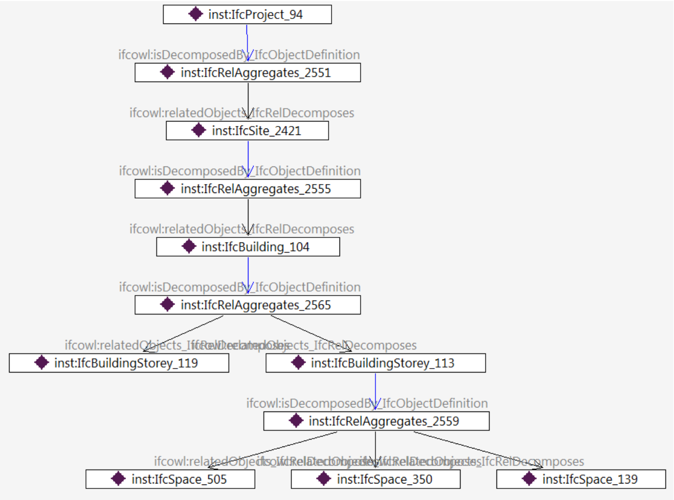

Figure 2. Spatial topology structure as stored in ifcOWL.

Many similar IfcRelationship instances (which represent the top property for IfcRelAggregates) exist in a typical IFC file. In our sample model, 233 of the 5580 instances are of type IfcRelationship (0 IfcRelAssigns, 19 IfcRelAssociates, 52 IfcRelConnects, 5 IfcRelDecomposes, and 157 IfcRelDefines). Indeed, most of the IfcRelationship instances connect specific building elements to particular IfcPropertySets (see Figure 3). Furthermore the IfcPropertySet, in turn, relates to a set of IfcPropertySingleValue instances, which in turn relate to IfcIdentifier and IfcBoolean instances that finally capture the actual property string and data elements ("isExternal true" in the case of Fig. 3).

Furthermore, 686 out of the 4222 class instances are of type list:OWLList. 417 of these OWLList instances are instances of ifcowl:IfcLengthMeasure_List. In fact, many of these lists relate to geometrical aspects, including IfcDirection and IfcCartesianPoint instances. An example of an IfcCartesianPoint instance and its lists is displayed in Fig. 4, showing three (!) lists to capture that a Cartesian point has coordinates (0,0,0). This clearly becomes even more verbose in the case of IfcPolyline entities and similar geometric objects that are defined by lists of Cartesian points.

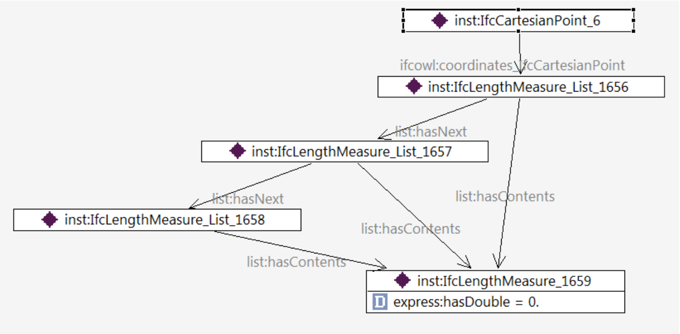

Figure 4. Definition of a Cartesian point.

As a last element that can be considered for simplification, is the choice to implement EXPRESS simple data types (BINARY, BOOLEAN, INTEGER, LOGICAL, NUMBER, REAL, STRING) using owl:Class wrappers. The choice to implement it as such in ifcOWL is inspired by the choice to do so in most, if not all, EXPRESS-to-OWL conversion efforts (see more detail in Pauwels and Terkaj (2016)). As an example, Figure 4 displays how expr:STRING instances are represented in our sample model. Note that 764 instances out of 4222 are actually such expr:STRING instances.

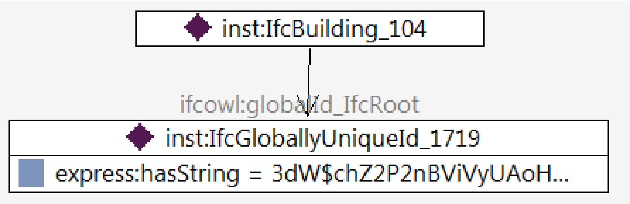

Figure 5. RDF graph representation for an expr:STRING instance related to an IfcBuilding instance.

4 THE SIMPLIFICATION PROCESS

4.1 *Implementation approaches*

Simplifying the considered graphs can be done in a number of ways. Of course, it is possible to simplify the graphs using procedural programming code. In this case, one could rely on one of the many software libraries available for handling RDF graphs and OWL ontologies. The advantage of this approach is that it can generate *any* kind of simplification and output.

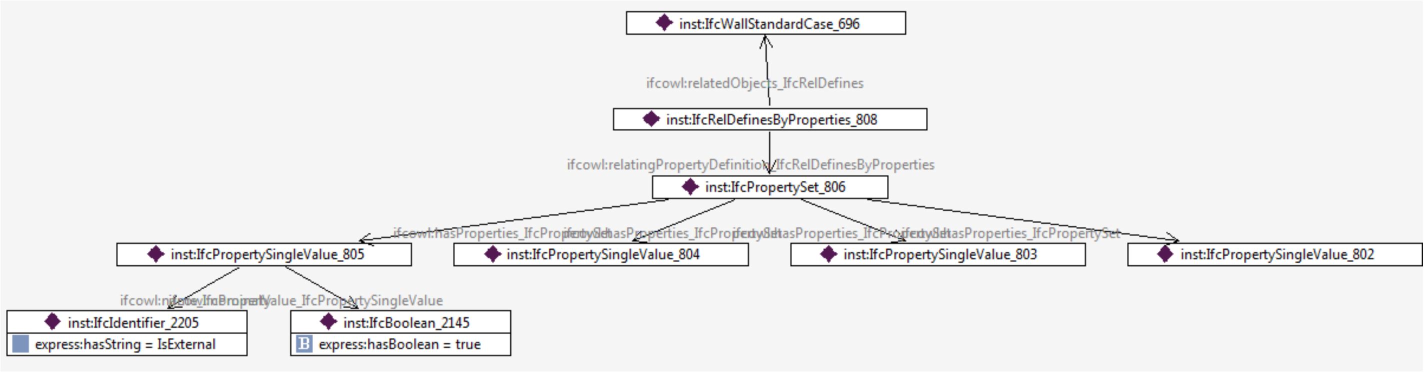

Figure 3. Structure used to relate properties to building elements.

Alternatively, however, one could also rely on the formal basis of the OWL language (Description Logics) and perform the simplification process in a declarative manner. Such an approach typically takes full advantage of query and rule languages commonly available for handling RDF data and OWL ontologies, like SPARQL and SWRL. These languages allow to declare sets of IF-THEN rules, either in SPARQL CONSTRUCT queries or in SWRL rules thus manipulating original considered data. A query or inference engine is then able to match the left hand side IF-parts of these rules and apply them to the available data (ifcOWL graphs), thus deducing the right hand side THEN-part of the rules.

Clearly, this second approach leads to a far more dynamic and on-demand simplification process. The data is natively stored in an ifcOWL graph pattern, and delivered to an end user interface in a simplified manner via such rules. Moreover, diverse different rule sets can be declared, simplifying the RDF data depending on the context in which the data is going to be used.

4.2 *Resulting simpleBIM data*

Regardless of the implementation approach taken, a number of relatively straightforward simplification suggestions can be considered here for our use case BIM model. Most of these suggestions relate directly to the presence of EXPRESS relics and features in the ifcOWL ontology. Among those we may cite the usage of data type wrapper classes and the usage of n-ary IfcRelationship instances. In our investigation, we implemented these suggestions in a simple parser and convertor that relies on the Jena software library for handling RDF graphs.

A first simplification that can easily be proposed in the context of a simplified usage of BIM data in a linked data context, is the release of geometrical and (re)presentation data. An IFC-SPF file of a BIM model contains a complete geometrical representation of a building. Such numerical information is seldom used in a linked data context. Clearly such information is far more intuitively handled in a 3D engine or authoring tool, rather than in a pure linked data context. So, it makes sense to remove this data from a simpleBIM building data cloud.

An easy way to make a clear split between parts of the IFC schema, is to use the architecture diagram that is listed for each schema to display resources and domains. In the case of IFC2X3, this diagram can be found in http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/order_by_architecture.htm. In our use case, we removed all instances that were part of the Presentation Resource, the Presentation Definition Resource, the Presentation Appearance Resource, the Profile Resource, the Representation Resource, the Topology Resource, the Geometry Resource, the Geometric Model Resource, and the Geometric Constraint Resource. This includes many of the complex statements, like the one displayed earlier in Fig. 4. After performing this operation, the RDF graph shrank to 6,927 triples (original: 10,173 triples) and 476kb (original: 767kb). This first simplification represents a reduction of the triple number and the file size by 31.9% and respectively 38%.

A second simplification rule relates to Fig. 5 and involves the 'unwrapping' of wrapped data types. Wrapping of data types is performed to allow a safe and uniform conversion of EXPRESS and IFC-SPF constructs into OWL and RDF respectively. In an instance file, however, these wrapped classed can often be unwrapped into explicit datatype properties. The example in Fig. 5 could thus be represented as displayed in Fig 6, using a custom datatype property simpleBIM:globalId that points towards an xsd:string. This can be done for all datatypes, including strings, integers, booleans, and so forth. After performing this operation, the RDF graph shrank to 3,897 triples (original: 10,173 triples) and 279kb (original: 767kb). We thus manage to lower the triple number by 43.74%, while reducing the file size by 41.39%, compared to the figures obtained in the previous step.

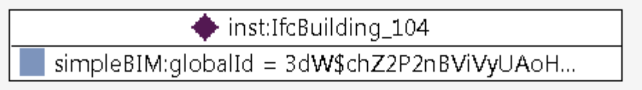

Figure 6. Simplified RDF graph for the expr:STRING example listed in Fig. 5.

A third simplification step involves the rewriting of properties, as it is done in (Mendes et al., 2015b). IfcOWL now handles property values and property sets as displayed in Figure 3, namely using a considerable number of intermediate steps (IfcPropertySet, IfcPropertySingleValue, IfcRelDefinesbyProperties). Although this might make a lot of sense in a BIM authoring tool, which typically uses a relational database with mapping tables between entities and tables of properties, this construct makes little to no sense in a linked data environment. In a linked data environment, a property is ideally immediately attached to an element. By rewriting this information as such, we can obtain a greatly simplified diagram as displayed in Figure 7. After performing this operation in our use case model, the RDF graph shrank to 1,630 triples (original: 10,173 triples) and 112kb (original: 767kb). We thus further reduce the number of triples, along with the file size (58.17% and respectively 58.86% less than in the previous step).

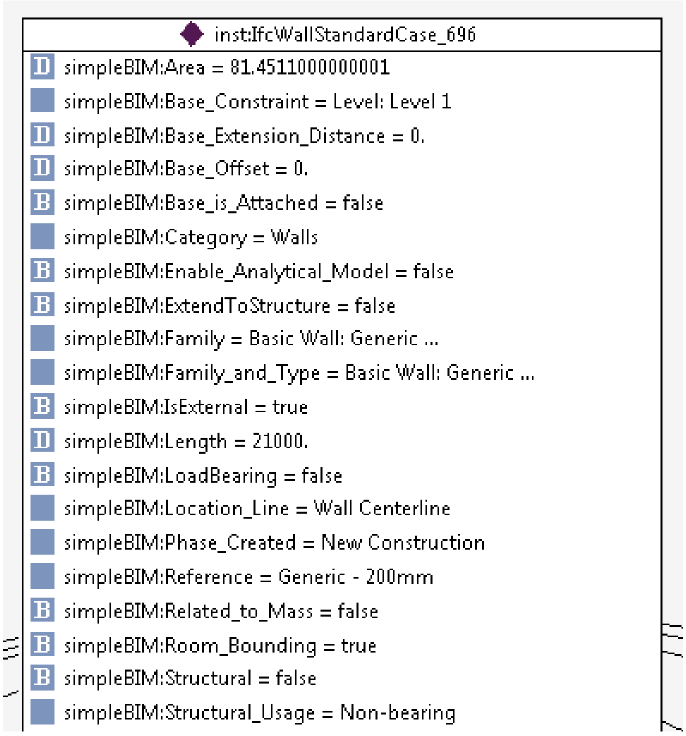
inst:IfcWallStandardCase_696
D simpleBIM:Area = 81.4511000000001
simpleBIM:Base_Constraint = Level: Level 1
D simpleBIM:Base_Extension_Distance = 0.
D simpleBIM:Base_Offset = 0.
B simpleBIM:Base_is_Attached = false
simpleBIM:Category = Walls
B simpleBIM:Enable_Analytical_Model = false
B simpleBIM:ExtendToStructure = false
simpleBIM:Family = Basic Wall: Generic ...
simpleBIM:Family_and_Type = Basic Wall: Generic ...
B simpleBIM:IsExternal = true
D simpleBIM:Length = 21000.
B simpleBIM:LoadBearing = false
simpleBIM:Location_Line = Wall Centerline
simpleBIM:Phase_Created = New Construction
simpleBIM:Reference = Generic - 200mm
B simpleBIM:Related_to_Mass = false
B simpleBIM:Room_Bounding = true
B simpleBIM:Structural = false
simpleBIM:Structural_Usage = Non-bearing

Figure 7: simpleBIM RDF graph with datatype properties directly associated to building elements.

A last simplification involves the replacement of the relational instances, which are all subclasses of IfcRelationship, with direct simpleBIM object properties between the applicable instances. This is already implemented in ifcWOD (Mendes et al., 2015b). Indeed, ifcWOD defines IfcRelationship and its related subtypes (e.g. IfcRelDefinesbyProperties) as OWL object properties. In ifcWOD, inverse attributes are also defined in order to link the IFC entities corresponding to related objects to the referring IfcRelationship entities. When applying this to SimpleBIM, we obtain the diagram displayed in Figure 8. After performing this operation in our use case model, the RDF graph shrank to 1,339 triples (original: 10,173 triples) and 83kb (original: 767kb). This represents a reduction of 17.85% for the number of triples and of 25.89% for the file size, when compared to the results achieved with the previous simplification.

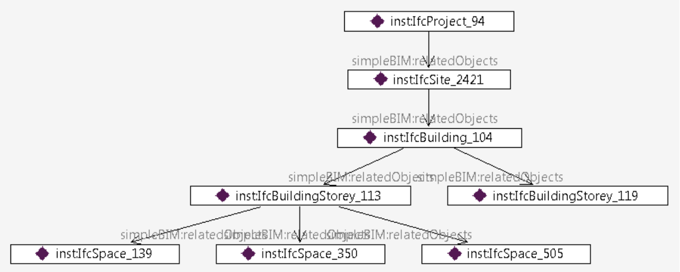

Figure 8. Simplified representation for the spatial structure of a building, as an alternative for the diagram in Fig. 2.

4.3 *Overall usability*

The same set of simplifications has also been tested for larger models, to display the significance of the size and complexity improvements that can be made, as well as to show general usability. The results of these tests are briefly outlined in Table 1.

Table 1. Simplification statistics for 2 reference models.

Model	***ifcOWL***	***ifcOWL***	***simpleBIM***	***simpleBIM***
1	767kb	10,173	83kb	1,339
Impr.	-	-	**89.18%**	**86.84%**
2	16,7 MB	225,135	1.029kb	16,836
Impr.	-	-	**93.98%**	**92.52%**

These results show an average size improvement of 89.68% in terms of triples (91.58% in terms of file size). This in itself shows the dramatic increase in usability of information. Furthermore, we wish to stress here that the actual RDF graphs are considerably simpler, while still representing the information that is typically used in a linked data context. Indeed, the eventually contained information comes close to the information that is typically described in non-IFC ontologies as documented in Section 2.

5 CONCLUSION

In the context of AEC, today's practical use of building information in form of IFC files has arisen several issues notably regarding the simplification of those files. When considering buildingSMART, a first approach for answering these issues comes in the form of IDMs and the related MVDs. Still, the underlying processes (e.g. IDM specification, MVD development) remain complex and time-consuming. Easy and intuitive ways to rapidly browse, query and use BIM information are not often available. The usage of semantic web technologies can remedy this situation, as these technologies allow to more dynamically manipulate the building information in the RDF graphs using web technologies, including query and rule languages.

In this paper, we have defined and exemplified 4 main approaches towards simplification of building models represented in ifcOWL graphs. While the first two of them are totally novel, the last two were adapted from the ifcWOD ontology (Mendes et al., 2015b). These simplifications can be dynamically applied on existing ifcOWL graphs. Average simplification percentages can be obtained of 89.68% in terms of triple count and 91.58% in terms of file size.

In principle, this method can easily be used to generate MVDs from full ifcOWL graphs. This responds to the idea of *generating* MVDs, in addition to merely specifying and validating Model Views, as outlined in Section 2.1. In this case, an MVD expressed in mvdXML should thus automatically be parsed, so that it informs the simplification process, and the output of the SimpleBIM process is an ifcOWL graph that contains only the information specified in the mvdXML graph. Note that the resulting graph in this case still needs to be valid ifcOWL, because an MVD is also required to result in valid IFC-SPF data. This leave out almost all of the outlined simplification possibilities.

However, additional and alternative simplification approaches can be considered as well, apart from the MVD generation phase, depending on the use case that is considered. For example, some use cases might prefer the usage of IfcPropertySet instances, which are now not considered in the simplification process. Other use cases might focus entirely on the geometry of the ifcOWL graph, a portion of information that is not retained in the presented examples. Hence, the proposed approach has a clear industrial value in the sense that it allows to intuitively supply IFC information in the custom form and custom size that is often required or demanded in the name of interoperability (even much more custom than what is available via MVDs).

The presented technique thus allows to adapt an RDF graph (in ifcOWL) into diverse alternative, less complex graphs, depending on the use case. Indeed, while ifcOWL is intended as a recommended standard, requiring close correspondence with the EXPRESS schema, the graph resulting from this SimpleBIM procedure is more of a usable extension of this standard. The same is true for the ifcWOD extension as proposed in Mendes et al. (2015b) and similar private industrial dynamic simplification mechanisms.

In terms of future work, we plan to investigate how the presented procedure can be used to transform and simplify ifcOWL graphs into the graphs used in the works proposed in Section 2.2: e.g. DogOnt (Bonino & Corno 2008), the OntoFM ontology (Dibley et al. 2012), and the sumo ontology (Niknam and Karshenas 2015). If such a procedure would be available, it means that ifcOWL information can be readily supplied in any of these ontologies, allowing a considerably improved information exchange process.

Furthermore, we intend to perform additional benchmarks, in order to highlight eventual advantages of SimpleBIM for SPARQL query simplification. We have to check that SPARQL queries defined using only SimpleBIM concepts are complete (in other words these queries do return all the expected results). We also want to examine the related query execution time, in order to identify if query simplification results in a sensible reduction of query execution time.

6 ACKNOWLEDGEMENTS

The authors wish to thank the Special Research Fund (BOF) of Ghent University for their generous support, and the French Agency of Research (ANR) for the financing of the BigSTEP project (funding ANR-16-MRSE-0024-01).

REFERENCES

Bonino, D. & Corno, F. 2008. DogOnt - ontology modeling for intelligent domotic environments. In *Proceedings of the International Semantic Web Conference*. Vol. 5318 of Lecture Notes in Computer Science (LNCS), 790-803.

BuildingSMART International 2015a. Overview page for the Linked Data Working Group. http://buildingsmart-tech.org/future/linked-data.

BuildingSMART International 2015b. ifcDoc Tool Summary. http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool.

Dibley, M., Li, H., Rezgui, Y. & Miles, J. 2012. An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction* 28: 1-14.

El-Diraby, T. E. 2013. Domain ontology for construction knowledge. *Journal of Construction Engineering and Management* 139(7): 768-784.

Karlshøj, J. 2011. Information Delivery Manuals. http://iug.buildingsmart.org/idms.

Kofler, M. J., Reinisch, C. & Kastner, W. 2012. A semantic representation of energy-related information in future smart homes. *Energy and Buildings* 47: 169-179.

Lima, C., Fies, B., Zarli, A., Bourdeau, M., Wetherill, M. & Rezgui, Y. 2002. Towards an IFC-enabled ontology for the Building and Construction Industry: the e-COGNOS approach. In *Proceedings of the eSMRT 2002 Conference*, 254-264.

Lima, C., El-Diraby, T. & Stephens, J. 2005. Ontology-based optimization of knowledge management in construction. *Journal of Information Technology in Construction* 10: 305-327.

Mendes de Farias, T., Roxin, A. & Nicolle, C. 2014. A Rule Based System for Semantical Enrichment of Building Information Exchange. In *Proceedings of the RuleML 2014 Challenge and the RuleML 2014 Doctoral Consortium hosted by the 8th International Web Rule Symposium (RuleML 2014)* Prague, Czech Republic, August 18-20, 2014, eds.: Theodore Patkos, Adam Wyner and Adrian Giurca. Aug 2014, Prague, Czech Republic. Vol-1211, pp.2. Available online at: http://publik.tuwien.ac.at/files/PubDat_231529.pdf

Mendes de Farias, T., Roxin, A. & Nicolle, C. 2015a. Semantic Web Technologies for Implementing Cost-effective and Interoperable Building Information Modeling. In *Proceedings of the 14th International Conference on Informatics in Economy (IE 2015)*.

Mendes de Farias, T., Roxin, A. & Nicolle, C. 2015b. IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties. In *Proceedings of the 32nd CIB W78 Conference on Information Technology in Construction*, 175-185.

Niknam, M. & Karshenas, S. 2015. Integrating distributed sources of information for con-struction cost estimating using Semantic Web and Semantic Web Service technologies. *Automation in Construction* 57: 222-238.

Pauwels P. 2015. IFC repository. http://smartlab1.elis.ugent.be:8889/IFC-repo/.

Pauwels, P. & Roxin, A. 2016. Additional files for ECPPM2016 Article. http://users.ugent.be/~pipauwel/ECPPM2016_SimpleBIM/index.html.

Pauwels, P. & Terkaj, W. 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction* 63: 100-133.

Reinisch, C., Kofler, M., Iglesias, F. & Kastner, W. 2011. ThinkHome Energy Efficiency in Future Smart Homes. *EURASIP Journal on Embedded Systems* 2011: 1-18.

Ruikar, D., Anumba, C., Duke, A., Carrillo, P. & Bouchlaghem, N. 2007. Using the semantic web for project information management. *Facilities* 25: 507-524.

Scherer, R. & Schapke, S.-E. 2011. A distributed multi-model-based Management Information System for simulation and decision making on construction projects, *Advanced Engineering Informatics* 25: 582-599.

Venugopal, M., Eastman, C. M., Sacks, R. & Teizer, J. 2012. Semantics of model views for information exchanges using the industry foundation class schema. *Advanced Engineering Informatics* 26(2): 411-428.

Weise, M. & Pauwels, P. 2015. Best practices for publishing and linking BIM data: scoping of IFC models. *Third International Workshop on Linked Data in Architecture and Construction*.