Conformance Checking of IFC Models with the Extensions in the MvdXML Checker

Muhammad Fahad[+], Nicolas Bus[*]
[+]Experis / ELAN IT
1240 Route des Dolines,
Valbonne 06560, France
email: {firstname.lastname}@experis-it.fr

Bruno Fies[*], Franck Andrieux[*]
[*]Centre Scientifique et Technique du Bâtiment
290 Route des Lucioles,
Sophia Antipolis 06904, France
email: {firstname.lastname}@cstb.fr

Abstract— **Building codes and standards are the rules and guidelines that specify the minimum acceptable level of safety for buildings. MvdXML Checker is a building code checker for the automatic verification of IFC models and to detect the non-conformities with the associated 3D visualization. The verification approach is based on MvdXML rules to be used within the MvdXML Checker or IfcDoc tool developed by the buildingSMART International to improve the consistent and computer-interpretable definition of Model View Definitions. In this paper, we propose many extensions in the MvdXML Checker which are very useful for the verification of IFC models and discuss its implement as a web service. After these extensions, still we analyze that this traditional approach of verification by the use of MvdXML is very limited and has narrow scope for the verification of IFC models. Major limitations are identified such as restricted scope of applying conditions and constraints on several branches of an IFC model, poor geometric analysis of an IFC model, lack of mathematical calculations, support of only static verification of a model, etc. Therefore, finally we present a need of an approach based on Semantic Web technologies that can easily be extended, configured and deployed for the dynamic and changing environment having broad spectrum of functionalities for the Verification of IFC models.**

Keywords- Conformance of IFC models; Building Code; MvdXML; BIM; Querying IFC models;

I. INTRODUCTION

Building codes are the rules and guidelines that specify the minimum acceptable level of safety, accessibility, general welfare, etc. of building models. Through building code and standards, organizations achieve their fundamental goal to protect public health, safety and general welfare as they relate to the construction and occupancy of buildings. The aim of validating Models is to align several specialized indexations of building components at both sides, assuming that they deal with the same abstract concepts or physical objects, but according to their separate representation prisms. Building Code is vital to detect non-compliance elements in the IFC model and to ensure its quality and reliability in the entire life-cycle of BIM [1], [2]. Since many years, *Industry Foundation Classes* (IFC) has been used by the *Architecture, Engineering and Construction* (AEC) industry for the building model representation. IFC is the complete and fully stable open and international standard for exchanging building data [3]. IFC has been designed to process a building's data model throughout its entire life cycle and to allow the inter-exchange of an information model without loss or distortion of data. Building SMART organization aims at publishing IFC and related buildingSMART data model standards. The buildingSMART data model standards are developed by the Model Support Group, and the implementation activities are coordinated by the Implementation Support Group [4]. Together, both groups organize the IFC software certification process. It aims to be a global standard for the BIM data exchanges.

To determine the quality of an IFC model after it has been developed and to fully automate quality assessment according to the French building code compliance regulations of IFC models is one of the goal of our enterprise, *Centre Scientifique et Technique du Bâtiment* (*CSTB*), through its research and development efforts. Building code compliance is a difficult job because it needs to check all building work must comply with the building codes (i.e., fire safety, accessibility, etc.). For this, verification each small IFC object must be tested to ensure its accuracy and validity along with test of overall IFC objects together in the IFC model. To achieve these goals, our research adopts a traditional approach using MvdXML [5]. The subset of the IFC schema needed to satisfy one or many Exchange Requirements of the AEC industry is called *Model View Definition* (MVD). The XML format used to publish the concepts and associated rules is MvdXML and it is regarded as an open standard [5]. It can be used with the IfcDoc tool [18] developed by the buildingSMART International to read and write MvdXML and to provide a graphical user interface for defining all content within MvdXML. It is aimed to improve the consistent and computer-interpretable definition of MVD as true subsets of the IFC Specification with the enhanced definition of concepts. MVDs provide additional rules for the IFC validation and focus on extracting integral model subsets for IFC implementation purposes. The buildingSMART is willing to support construction domain developers in reusing its leading openBIM standard IFC as a baseline to set up specific data exchange protocols to satisfy exchange requirements in the industry. The buildingSMART International has developed IfcDoc tool for creating Model View Definitions. Based on the newly developed MvdXML standard, just Model View Definitions can now be easily

 ISBN: 978-1-61208-651-4

developed using the IfcDoc tool. The tool and methodology can be applied to all IFC releases (IFC2x3, IFC4, etc.). For the validation of an IFC file against a particular model view, IfcDoc tool user interface displays a pane on the right side containing object instances within the file matching definitions selected in the tree view. The end-user can generate a report in the HTML format indicating if the file is valid according to the specified model view, and detailing what passes or fails. IfcDoc tool and conformance checking by MvdXML technology is a good candidate for the simple rules on small IFC models. Although it fulfills lots of requirements for the code compliance of IFC models but still there is a gap which needs to be fulfilled. We took this opportunity to fulfill this gap and propose certain extensions. Finally, in this paper, we suggest going beyond MvdXML and present a need of an approach based on Semantic Web technologies that can easily be extended, configured and deployed for the dynamic and changing environment having broad spectrum of functionalities for the Verification of IFC models.

The rest of paper is organized as follows. Section 2 presents related work. Section 3 discusses a usecase and extensions in the MvdXML for the conformance checking of IFC models. Section 4 points several limitations of MvdXML validation rules and discusses to go beyond MvdXML. Section 5 concludes this paper.

II. RELATED WORK

Code Compliance checking is targeted by the industry and researchers in order to provide the facilities to stake-holders like the delivery of high quality IFC model to ensure more accurate, consistent and reliable results in the life-cycle of BIM. In the BIM based research literature, there are three ways for the conformance checking of IFC models as discussed by Pauwels and Zhang [16]. First, we have the '*hard coded rule checking'* where rules are integrated inside the application. This approach is adopted by Solibri Model Checker [17] and IfcDoc tool [18] along with the MvdXML rules. The second approach is '*rule checking by querying'* the IFC model followed by K.R. Bouzidi et al [19]. In this approach, BIM is interrogated by rules, which are formalized directly into SPARQL queries. The third is a semantic rule checking approach with dedicated rule languages (such as SWRL [7], Jess [24] or N3Logic [25]) adopted by H. Wicaksono et al. [20] based on SWRL rules, Pauwels et al. [21] based on N3Logic rules and M. Kadolsky et al. [22].

Besides these researches, recent years revealed some contributions based on Semantic Web technologies. SWOP-PMO project is one of recent contributions that use formal methodology based on the Semantic Web standards and technologies [14]. It uses OWL/RDF to represent the knowledge, and SPARQL [8] queries and Rule Interchange Format (RIF) to represent the rules. The RDF/OWL representation is not derived from the written knowledge but has to be remodeled in accordance with the rules of OWL/RDF. There are some other works for the semantic enrichment of ontologies in the construction and building domain. Emani et al. proposed an ontology-based framework for generating an OWL Description Logic (DL) [23] expression of a given concept from its natural language definition automatically [15]. Their framework also takes into account an IFC ontology and the resultant DL expression is built by using the existing IFC entities. To enable the compliance checking of the repository through the digital building model, Fahad et al. have contributed a framework for mapping certification rules over BIM [11]. They aimed to align several specialized indexations of building components at both sides, by extending IfcOWL ontology with bSDD vocabulary (i.e., synonyms and description) as enriched IfcOWL ontology to deal with the same abstract concepts or physical objects. Fahad et al. also investigated semantic web approach by using SWRL and traditional approach by the use of IfcDoc tool and analyzed that the semantic web technique represent more global scope with larger visibility of querying for the validation of IFC models [12].

III. EXTENSIONS IN THE MVDXML CHECKER

An MvdXML document contains an instance of mvd:MvdXML as a main element, which defines a set of reusable concept templates, mvd:ConceptTemplate, and a set of model view definitions, mvd:ModelView. Each model view defines an applicable schema like IFC2X3 or IFC4 and may set a link to a base view definition if it is an add-on view. The validation of IFC building models is vital in the BIM-based collaboration processes and can be done via IfcDoc tool. Using IfcDoc tool with the MvdXML rules, an end-user performs three steps of automatic control sequence. The IfcDoc engine loads an IFC file and MvdXML files containing rules, and then it executes defined rules over IFC model. Finally, it generates a report indicating compliance (compliant/non-compliant) of each item under the rule. It assigns each rule a green or red depending on whether the item is/is-not in compliance to the defined rules. Besides IfcDoc tool, an end-user can use standalone MvdXML Checker (a java-based research prototype) to meet the requirements of verification models. Working on different usecases, we found that we need extensions in the MvdXML Checker to meet the requirements of real world scenarios. The following subsections present motivation scenario and the extensions we made in the MvdXML checker.

A. UseCase Scenario – conditions and constraints

When we need to access the name/label of an IFCSpace, we can simply access the name attribute of the IFC schema. This is a very simple case which can be employed very easily. But there are some scenarios which we cannot implement with the current implementation of MvdXML. During the validation process we can have conditions and constraints in the rule for certain scenarios which is not supported by the MvdXML Checker. According to IAI, "*An IFCConstraint is used to define a constraint or limiting value or boundary condition that may be applied to an* *object or to the value of a property*". This element is defined within the elements mvd:EntityRule and mvd:AttributeRule and represents a restriction on an attribute, which may require the value, type, or collection size to have equality (or other comparison) to a literal value or referenced value. For example consider a scenario illustrated in Figure 1, where we can apply conditions and constraints on the IFCSpace. Figure 2 demonstrates the MvdXML of this small example. One can note that in MvdXML rule file, description="*" in the rule description demonstrates the presence of a condition and its absence represents the constraints. There can be very complex situation, for example, as depicted in Figure 3, a chain of hierarchy, where conditions and constraints are involved. Therefore, precise tackling of conditions and constraints should be analyzed and integrated properly in the MvdXML Checker.

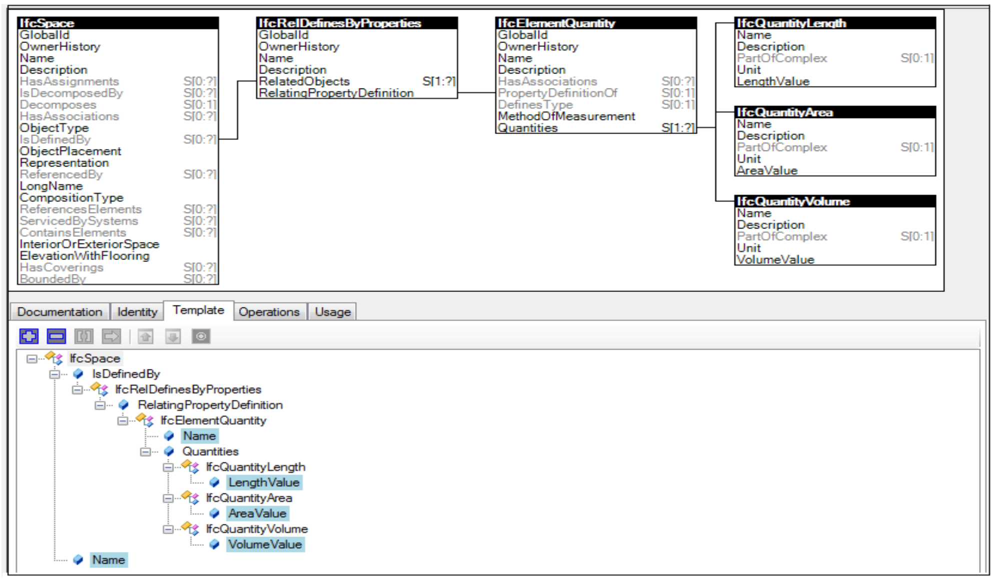

Figure 1. IFCSpace with various attributes

```xml
    <ConceptTemplate uuid="f4a3a95e-fa59-41a7-83ad-5cff467b25bc" name="test" applicableSchema="IFC4" applicableEntity="IfcSpace">
      <Rules>
        <AttributeRule AttributeName="IsDefinedBy">
          <EntityRules>
            <EntityRule EntityName="IfcRelDefinesByProperties">
              <AttributeRules>
                <AttributeRule AttributeName="RelatingPropertyDefinition">
                  <EntityRules>
                    <EntityRule EntityName="IfcElementQuantity">
                      <AttributeRules>
                        <AttributeRule RuleID="name" Description="*" AttributeName="Name" />
                        <AttributeRule AttributeName="Quantities">
                          <EntityRules>
                            <EntityRule EntityName="IfcQuantityLength">
                              <AttributeRules>
                                <AttributeRule RuleID="length_value" Description="*" AttributeName="LengthValue" />
                              </AttributeRules>
                            </EntityRule>
                            <EntityRule EntityName="IfcQuantityArea">
                              <AttributeRules>
                                <AttributeRule RuleID="area_value" AttributeName="AreaValue" />
                              </AttributeRules>
                            </EntityRule>
                            <EntityRule EntityName="IfcQuantityVolume">
                              <AttributeRules>
                                <AttributeRule RuleID="volume_value" AttributeName="VolumeValue" />
                              </AttributeRules>
                            </EntityRule>
                          </EntityRules>
                        </AttributeRule>
                      </AttributeRules>
                    </EntityRule>
                  </EntityRules>
                </AttributeRule>
              </AttributeRules>
            </EntityRule>
          </EntityRules>
        </AttributeRule>
        <AttributeRule RuleID="name" Description="*" AttributeName="Name" />
      </Rules>
    </ConceptTemplate>
  </SubTemplates>
</ConceptTemplate>
```

Figure 2. MvdXML showing conditions and contraints

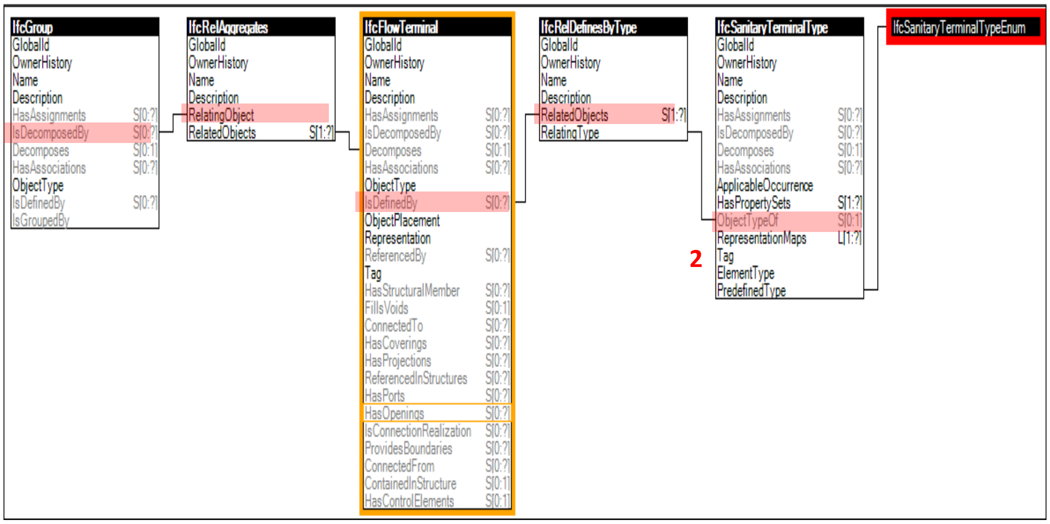

Figure 3. Applying conditions and constraints on the Attributes of an IFCSpace concept

Figure 4. Interface of output by the Extended MvdXML checker

B. Extentioned MvdXML Checker as a WebService

We have implemented several extensions including the use case defined above in the MvdXML checker and developed a web service [26] so that it can be used by end-users. First end-users upload an IFC model and choose to apply code compliance rules over their IFC model. Then, the checker executes control rules to verify chosen protocols on the input model by starting a web service. Web service executes the control rules and displays the results as an output in the html form as illustrated in Figure 4. Each of the rule that detects non-compliant elements are highlighted with the red color so that end-users would be alert in those particular cases. When one clicks on the red highlighted rule, the web service loads non-compliant elements with their *Guids, Names and Types*.

IV. NEED OF NEW IMPLEMENTATIONS FOR THE COMPLIANCE CHECKING OF IFC MODELS

This section highlights various limitations of MvdXML Checker and then discusses a real world use case scenario of verification rule which cannot be modeled by using MvdXML. Later in this section, we propose a semantic based solution for the building code conformance checking of IFC models.

A. Limitations of MvdXML Checker

Complex nature of IFC makes the information retrieval difficult and as a consequence it affects the validation process of MvdXML rules. Many tasks for an IFC model, such as information retrieval, model validation, etc., do not achieve real-time performance in the real-world BIM scenarios. There are many drawbacks of MvdXML for extracting building views such as: lack of logical formalisms, solely consideration of IFC schema and MVD-based view constructors are not very flexible and dynamic [6]. Verification by MvdXML rules are also very limited. Major limitations are identified such as restricted scope of applying conditions and constraints on several branches of an IFC model, poor geometric analysis of an IFC model, lack of mathematical calculations, support of only static verification of a model, etc. It does not show the cause or provide mechanisms for reasoning the inconsistencies or anomalies. However, on the other hand, Semantic Web technologies, especially SWRL or SPARQL, allow for the semantic verification of IFC models to enable the compliance checking of IFC construction models with fast querying performance. The next section will show motivation for the semantic based verification of IFC models.

B. Need of New Semantic Implementations

While working with the MvdXML and IFC tools, we realized that there is no support to build new concept and/or high level vocabulary dynamically or create a new rule using existing concepts. For example, a simple rule that is based on *'Highest Storey'* would neither be possible with IfcDoc tool nor with the MVDXML specification. But, if we process an IfcModel and build a semantic repository with the geometry information and materialize high level vocabulary via SPARQL Rules and Queries then we can build verification rules over the *Highest Storey*. With the help of Sparql rules on the geometry data (i.e., minimum and maximum values of X, Y and Z coordinates) of IFC objects, we can infer elements which are *Above or Below* with respect to each other. Once we can infer IFC objects then *"Not Exists {B above A}"* concludes A as a *Highest Storey* having nothing over it. Figure 5 illustrates this scenario.

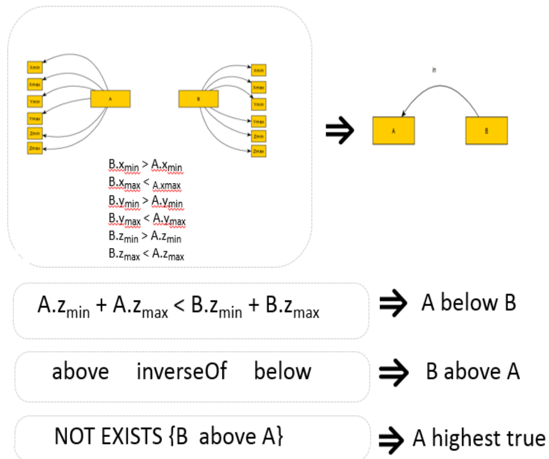

Figure 5. Semantic illustration of Highest Storey Concept using geometry data from an IFC model

Besides building verification rules over geometry data, querying semantic model is faster and gives a good run-time. One can customize queries easily and according to requirements. Using MvdXML, there is no intermediate state and IfcDoc tool gives no explanation for the reason of non-compliance. On the other hand, the Semantic Web technology is a good compromise between development efforts and opportunities. The graphical representation of RDF allows rules to be more intuitive and more efficient to reason and execute. As SWRL [7] and SPARQL [8] are W3C recommendations, a lot more functionalities are added to meet the requirements of the real world scenarios. For example, one can perform calculations in SWRL, which we cannot do in MvdXML. In addition, we can also define new attributes and elements, evolve the existing values, and give them values based on the initial axioms in the repository and store them back in our repository for further processing. In addition, many complex verification rules execute on the combination of IFC objects or depend on small rules, but this is not much flexible and most of the time concatenation of conditions and constraints is not possible in the IfcDoc using MVDXML specification.

C. Work in Progress – Semantic based Verification

To meet the requirement of semantic checking, we have implemented semantic based approach for the building code compliance. We have used IFC-to-RDF-Converter developed by Pauwels and Oraskari [13] to get a semantic repository (RDF triplets) [10] equivalent from an IFC model. We apply filtration to get an RDF equivalent compact triplet file to avoid several IFC elements, such as Person, Address, Material-List, etc. Then, we extract all geometry data from the input IFC model by using BIM-Server plugins [27]. BIM-Server is an open source toolkit to work with the IFC models and provides two plugins (i.e., *IFCOpenShell* and *IFC Engine DLL*) for the extraction of geometry data. Once we gather all triplets (i.e., Filtered RDF file of IFC model and Triplets of Geometry data), we load them into the Stardog triple store [9] for the fast querying, searching, and analyzing of RDF triplets. Over these triplets, we build our high level vocabulary by using SPARQL rules (e.g., highest storey concept explained above). Finally, we formalized verification rules into SPARQL queries which bring triplets (i.e., non-conformance elements) in the case of non-compliance of building code. The whole architecture is illustrated in Figure 6. We build several test cases comprise of different queries on different sizes of IFC models by the traditional approach via IfcDoc and the ontology-based semantic approach via SPARQL. From the initial results, we conclude that SPARQL queries are flexible for retrieving data and do the validation in an optimized way giving better run-time as compared to the traditional approach via IfcDoc. But the conversion from IFC to RDF and then storage of triples into stardog takes time. But, once the stardog triple store is loaded with the data, it is much faster querying and validation of IFC models. SPARQL queries can be modified easily with the new or customized conditions and constraints for the conformance checking against the triple store. Besides flexibility, reasoning is another advantage of Semantic Web technology, as the IfcDoc tool does not provide any justification. With queries and rules, we can identify reasons of inconsistencies and anomalies via RDF graph traversals.

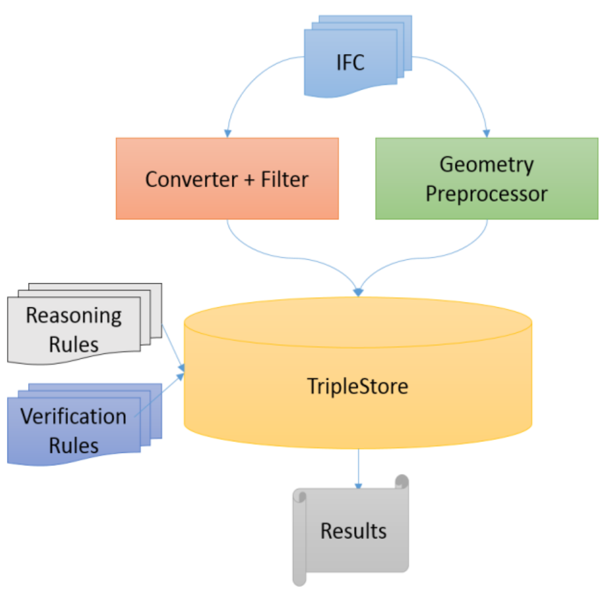

Figure 6. Architecture of semantic-based Approach

V. CONCLUSION AND FUTURE WORK

An IFC is a specific data format that aims at allowing the inter-exchange of an information model without loss or distortion of data. Building Code compliance is vital to ensure the quality and reliability of an IFC model. Building code compliance of IFC models is a hot issue for the researchers. One of the ways to apply code compliance of IFC files is by the use of MvdXML checker. However, current implementation needs further extensions so that all the rule specifications can be covered and tested. In this paper, we presented extensions we made in the implementation of MvdXML Checker available for the validation of IFC models. The proposed extensions were implemented in the form of a web service. End-users can invoke this service to check code-compliance according to their own rules over their desired IFC models. Finally, this paper also addresses the need of a semantic approach towards the automatic verification requirements to warn the non-conformities as a hot challenge. Our on-going research work is to develop and investigate a semantic web approach. We transform an IFC model into a RDF model and query on RDF triplets to meet the requirements of code compliance. It is obvious that only a well-engineered IFC model that passes code compliance can serve best in the entire life-cycle of BIM.

REFERENCES

[1] V. Rebekka, J. Stengel, and F. Schultmann, "Building Information Modeling (BIM) for existing buildings—Literature review and future needs," Automation in construction, vol. 38, pp. 109-127, 2014, doi: 10.1016/j.autcon.2013.10.023

[2] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, "BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors," Hoboken, New Jersey, Wiley, 2008.

[3] V. Thein, Industry Foundation Classes (IFC), "BIM Interoperability Through a Vendor-Independent File Format," A Bentley White Paper, September 2011.

[4] Building Smart, International home of openBIM, https://www.buildingsmart.org/standards/ [last access: june 2018]

[5] T. Chipman, T. Liebich, and M. Weise, "MvdXML specification of a standardized format to define and exchange MVD with exchange requirements and validation Rules," version 1.1 Final, Febrary 2016. Available at: http://www.buildingsmart-tech.org/downloads/mvdxml /mvdxml-1.1/final/mvdxml-1-1-documentation

[6] T. Mendes de Farias, A. Roxin, and C. Nicolle, "A Semantic Web Approach for defining Building Views," buildingSMART Summit Jeju, Korea, 28 Sept, 2016.

[7] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," 2004.

[8] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/, 2008. [last access: June 2018]

[9] Stardog, http://stardog.com/ [last access: June 2018]

[10] D. Brickley, R.V. Guha, and B. McBride, (2004). "RDF vocabulary description language 1.0: RDF Schema," W3C Recommendation 2004. [last access: June 2017]

[11] M. Fahad, Nicolas Bus., and F. Andrieux, "Towards Mapping Certification Rules over BIM," Proc. of the 33rd CIB W78 Conference, Brisbane, Australia, 2016.

[12] M. Fahad, N. Bus, F. Andrieux, "Towards Validation of IFC Models with IfcDoc and SWRL," In proc of The Twelfth International

Conference on Internet and Web Applications and Services (ICIW), italy. pp. 7-13, 2017.

[13] P. Pauwels and J. Oraskari, "IFC-to-RDF Converter" https://github.com/IDLabResearch/IFC-to-RDF-converter, [last access: June 2017]

[14] F. Josefiak, H. Bohms, P. Bonsma, and M. Bourdeau, "Semantic product modelling with SWOP's PMO, eWork and eBusiness in AEC," pp. 95-104

[15] C.K. Emani, C. Ferreira Da Silva, B. Fiès, P. Ghodous, and M. Bourdeau., "Automated Semantic Enrichment of Ontologies in the Construction Domain," Proc. of the 32nd CIB W78 Conference, Netherlands, 2015.

[16] P. Pauwels and S. Zhang, "Semantic Rule-Checking for regulation compliance checking: An overview of strategies and approaches," Proc. of the 32nd CIB W78 Conference, Netherlands, 2015.

[17] L. Khemlani, "Solibri model checker," AECbytes Product Review March 31, 2009

[18] IfcDoc Tool, available at: http://www.buildingsmart-tech.org/specifications/ specification-tools /IfcDoc-tool/IfcDoc-help-page-section/IfcDoc.pdf, 2012. [last access: June 2017]

[19] K. R., Bouzidi, B., Fies, C., Faron-Zucker, A. Zarli, and N. Le Thanh, "Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry," Future Internet, Special Issue Semantic Interoperability and Knowledge Building, 4 (3). pp. 830-851, 2012.

[20] H. Wicaksono, P., Dobreva, P. Häfner, and S. Rogalski, "Ontology development towards expressive and reasoning-enabled building information model for an intelligent energy management system," Proc. of the 5th KEOD, pp. 38-47. SciTePress, 2013.

[21] P. Pauwels, D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, R. Van de Walle, and J. Van Campenhout, "A semantic rule checking environment for building performance checking," Automation in Construction. 20 (5). pp. 506-518, 2011.

[22] M. Kadolsky, K. Baumgärtel, and R. J. Scherer, "An ontology framework for rule-based inspection of eeBIM-systems," Procedia Engineering. vol. 85, 2014, pp. 293-301, doi:10.1016/j.proeng.2014.10.554.

[23] OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11 December 2012, https://www.w3.org/TR/owl2-overview/ [last access: June 2018]

[24] E. Friedman-Hill, "Jess in Action: Rule Based Systems in Java," Manning Publications. ISBN 1-930110-89-8, 2003

[25] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler, "N3Logic: A logical framework for the World Wide Web," Theory and Practice of Logic Programming. vol. 8 (3), 2008, pp. 249-269, doi:10.1017/S1471068407003213.

[26] Extended MvdXML Checker as as web service: https://svc-bim-checker.dev.coplus.fr/CheckerService/v2/ui

[27] Open source BIMserver-plugin available at: https://github.com/opensourceBIM/IfcOpenShell- BIMserver-plugin [last access: June 2018]