Methodologies for requirement checking on building models

Methodologies for requirement checking on building models

A technology overview

T. Krijnen[1] and L.A.H.M. van Berlo[2]
[1]Eindhoven University of Technology
Eindhoven, The Netherlands
t.f.krijnen@tue.nl
[2]Netherlands organisation for applied scientific research TNO
Delft, The Netherlands
leon.vanberlo@tno.nl

Key words: Building Information Modelling, data management, code compliance, automation, model checking

Abstract: The use of Building Information Modelling (BIM) has increased in the Architectural and Urban domain. Stakeholders within distinct disciplines collaborate and exchange such information models digitally. In order to strive for an interoperable use of the models, requirement documents are being written by stakeholders, standardisation bodies and governments. Such documents pose additional requirements to the exchange of building model definitions and limit the scope of information to something that is relevant to the disciplines the exchange pertains to, the phase of the construction project and the level of development of the project. For effective collaboration processes, checking these requirements in an automated and unambiguous way is of crucial importance. Yet, requirement definitions often comprise natural language texts and academic and commercial tools being developed in this regard are fragmented and heterogeneous. Furthermore, the models being checked are of uncertain quality because the semantics of the schema are not rigorously formalized and enforced and models contain redundancies that affect their reliability. This paper urges for more developed schema semantics and illustrates how the body of technical means, such as classification system, concept libraries, query languages, reasoners and model view definitions are related to one another and to the concept of automated rule checking.

1. INTRODUCTION

1.1 Current practice

In recent years the use of Building Information Modelling (BIM) has increased in the Architectural and Urban domain. BIM has become an umbrella term for many recent innovations and insights based on data, but primarily signifies the transition from 2D CAD drawings to semantically rich information models in the building industry. Novel use cases have been enabled by virtue of the added semantic information.

In order to guarantee an interoperable use of these models, requirement documents are being written that make statements on the validity or appropriateness of (parts of) building model definitions. These use cases range from checking the conformance to rules set out by (local) governments to in-house guidelines and Level Of Development (LOD) specifications. In a similar fashion, the compliance to building codes needs to be checked in order to obtain permits. Several terms have been coined to name such requirements, such as “BIM norms”, “BIM employers requirements” and “BIM protocols”. They govern the requirements that people, organisations and government pose on the data or edifice that is delivered to them. Being able to check these requirements in an automated way is highly desirable for effective data exchange and high quality end-results.

On a technical level, various approaches are being researched to assess these requirements on actual models. These approaches allow for different levels of expressivity and extents to which they allow for automation, modularity and reuse. This paper will give an overview of available technical solutions to automate data requirements checking. It will state characteristics, advantages and limitations pertaining to these technologies. The technologies included in this overview are the IFC schema and its implementers agreements, Model View Definitions (mvdXML), classification systems and concept libraries, query languages, reasoners and proprietary software solutions.

1.2 State of the art

Various model checking platforms exist and are described in literature or are commercially available. A platform used in practice is *Solibri Model Checker (SMC)*. It is a JAVA-based executable that reads an IFC model and provides proprietary processing routines to facilitate the rule checking on common operations in the Architecture Engineering and Construction

(AEC) industry. This includes checking for the existence of attributes (so called pre-checking), and more advanced fire exit and evacuation checking and comparisons against the schedules and design briefs. SMC is a proprietary application that implements rule by means of hard-coding them in program code, therefore, other than adjusting specifically targeted parameters, this program is not extensible to add new types of rules.

Jotne EDModelChecker (EDM) is a commercial library that separates the definition of rules from the program code. Rules are defined in the EXPRESS modelling language, an ISO certified open standard, in which the IFC schema is conceived as well (Eastman, et al., 2009).

FORNAX is among the first large government-involved effort towards automated rule checking in the building industry. It is part of the Singapore CORENET platform, developed as an automated system to regulate building permits. It is implemented on top of the EDM Model Checker (Khemiani, 2005).

An attempt to aid the formalization process of rule definitions is provided by *SMARTcodes*, which presents methods of converting codes and standards from textual natural language definitions into computer code. This is accomplished by means of semantically structured domain knowledge (Nawari, 2012).

Automated approaches to validate the conformance of a model to a Model View Definition (MVD) are described in (Zhang, et al., 2013). But as described in (Solihin, et al., 2015), what constitutes a valid, meaningful and unambiguous exchange is broader than what currently can be expressed in such a MVD and includes in addition aspects such as geometrical and topological correctness, for example that spaces are correctly bounded and that the faces that constitute this boundary conform to their (typically planar) underlying surface geometry.

Disambiguation is a crucial part of the formalization of rule definitions. For this purpose the use of multilingual concept libraries is crucial as it allows to unambiguously point to a well understood concept from within diverse national classification systems and different languages (Palos et al., 2014). In the AEC industry initiatives are being undertaken to implement such concept libraries specific to the industry, such as the buildingSMART Data Dictionaries (bsDD). In addition, well established ontologies with a broader scope are available, for example (Navigli and Ponzetto, 2012) and (Miller, 1995).

An orthogonal line of research is initiated by (Krijnen and Tamke, 2015) that tries to employ machine learning concepts, such as anomaly detection to enable model checking without the à priori definition of formal rules, but instead deduce a norm to which most building elements conform to and flag the elements that deviate from this norm.

(Eastman, et al., 2009) describes a variety of rules and discusses the implementation of them in various of the available rule checking platforms. Often these rule sets are a mixture of safety or programmatic requirements. This work identifies major issues, such as a lack of extensibility of some of the platforms and resonates the finding that none of the platforms address the entire scope of rule checking. This entails the process of converting rules from natural languages into formal definitions, pre-checking the suitability of the model for more rigorous checking, executing the checks that often need geometrical and topological functionality to abstract building models into spatial structures suitable for e.g. fire exit checking and finally reporting the results.

1.3 Problem statement

The authors observe that on the one hand there is a wide variety of rules being developed, predominantly in natural languages within government bodies and standardization institutes. Yet, there is no easily apprehensible overview of the implications on computational complexity and decidability for such rules in the context of automated rule checking. On the other hand there is a wide variety of technical research directions undertaken that try to solve the notion of automated rule checking on top of various platforms with different levels of expressivity and with different requirements and prerequisites for the definition of rules. Often these technical ventures are focussed on isolated parts and not on the overall process. A general overview that connects and harmonizes the various approaches in this field seems to be lacking and is trying to be created in this paper. For that purpose an allegorical example of a rule is approached from various technical means. From this analysis the complimentary nature of these technical means can be seen.

2. DEFINITION OF A RULE

As an allegorical example of a rule, to be used throughout this paper, a seemingly simple example has been chosen that touches on and interconnects the relevant technical means of rule checking described in this paper. The rule is given below in Listing 1.

Listing 1. Rule introduced in this paper

```
A blind wall should not be longer than 3 meters
```

2.1 Relevance of concept libraries

In the particular case of the rule in Listing 1 three concepts can be identified: “length”, “meter” and, most interestingly, “blind wall”. Ontologies such as the Getty Architecture and Art Thesaurus (AAT) readily contain an identifiable concept that can be used for this purpose (Krijnen and Beetz, 2016), while there is no explicit semantics in the IFC schema to annotate walls of this type. In general, requirements, like the example above, are provided in natural language and the process of reinterpreting them into something formally understood, and to include the implicitly available domain knowledge, is a significant part of coming to an automated approach to checking such rules (Sohilin and Eastman, 2015). After understanding the formal implications of the rule, the process of rule checking relies on disambiguating terms in the equation by means of classification systems or concept libraries (Eastman et al., 2009). For a “blind wall” in AAT the following definition is given provided in Listing 2.

Listing 2. Definition of a blind wall in the Getty Architecture and Art Thesaurus

Walls whose whole surfaces are unbroken by windows, doors, or other openings.

With this definition the rule can be formalized as given in Listing 3, which is to say that a valid wall implies that either its length is less than or equal to three meters or no openings are associated to the wall element.

Listing 3. Formalization using predicate logic of the rule in Listing 1.

$$\textbf{valid\_wall}(\mathcal{W}) \rightarrow (\textbf{length}(\mathcal{W}) \leq 3\text{m}) \vee (\exists \mathcal{O}\text{: } \textbf{opening}(\mathcal{W}, \mathcal{O}))$$

2.2 Relevance of query languages

After the formalization, the rule can be encoded as a query. For the purpose of reporting issues with model, that is to say, return elements within a model that violate the rule, the query in Listing 4 has been negated. Only the second part of the conjunction is included initially. In this particular case the SPARQL query language has been used. This necessitates that the building model definition in IFC has to be converted to a linked data representation, for example as defined by (Pauwels and Terkaj, 2016). The query returns unique identifiers for walls to which no IfcRelVoidsElements are related. This is the objectified relationship in the IFC schema that connects elements to openings, which in turn are connected to windows and doors.

Listing 4. SPARQL query to return walls without openings

```
PREFIX ifcowl: <http://www.buildingsmart-tech.org/ifcOWL#>

SELECT ?wall_guid_value
WHERE
{
   ?wall a ifcowl:IfcWallStandardCase ;
      ifcowl:GlobalId ?wall_guid .

   ?wall_guid ifcowl:has_string ?wall_guid_value .

   FILTER NOT EXISTS {
      ?rel a ifcowl:IfcRelVoidsElement ;
         ifcowl:RelatingBuildingElement_of_IfcRelVoidsElement ?wall .
   }
}
```

However, when the results of such a query on a commonly used example model[1] are then visualized, the set of elements returned consists of several unexpected elements. Elements identified by this query do not conform what would be typically understood as a wall. As has also been noted in (Krijnen and Tamke, 2015) models might contain flaws pertaining to misclassifications into element categories.

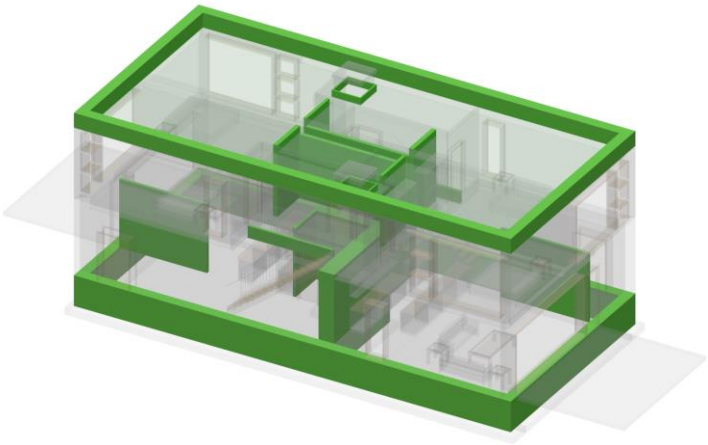

Figure 1. Elements returned by the query in Listing 1 include elements not typically considered as walls. Elements that are matched by the query are printed in solid green. Unmatched elements are printed transparently.

[1] http://www.nibs.org/?page=bsa_commonbimfiles

2.3 Relevance of schema semantics

As illustrated in Figure 1, many of the elements returned by the query to retrieve blind walls, while they indeed do not contain openings, also do not appear to match the common idea of a wall. According to the IFC standard, which draws from ISO 6707-1 for most of its terminology, a wall is a "vertical construction usually in masonry or in concrete which bounds or subdivides a construction works". However, as has been noted by (Amor, 2015), with the IFC schema expanding its size and domain coverage, the body of codified and formal where rules is not growing along. The only semantic constraint to a wall (IfcWallStandardCase) is that there should be a single material layer set defined that depicts the distribution of material over its cross section. There is no formal apprehension in place of the fact that elements, which do not function as a wall in the model, should not be classified as such.

An additional issue with the body of formal rules defined in the IFC standard is that these rules are defined in the EXPRESS language, which is a relatively old specification for which little support is available in terms of its rule language. This calls for mapping such rules to other formal languages, for example as implemented in (Terkaj and Šojić, 2015).

2.4 Relevance of Model View Definitions

Model View Definitions specify additional constraints on the validity of an exchange of information. Such an MVD then describes the subsets of the schema that supports the needs of a particular data exchange (Lee et al., 2016). The language to describe such MVDs, mvdXML, is not as expressive as other rule languages, making it relatively easy to implement in software and making the definition of such rules straightforward. On the other hand it necessitates that if higher levels of expressiveness are required, other languages need to be incorporated, for example SWRL in (Lee et al., 2016). In addition, by specifying what information is relevant, they also define where is to be found in the model.

The left-hand side of the conjunction in the query in Listing 3 constraints the maximum length of a blind wall. IFC allows the definition of key-value pairs by means of its IfcPropertySets and IfcElementQuantities. They form an extensible means to encode information that is not directly prescribed in the IFC schema. The extensibility stems from the usage of free-form strings as the keys of the pairs, which is the sole constituent that defines the meaning. This is also why additional conventions are needed to guarantee an interoperable interpretation of the information conveyed in these key-value

pairs. MVDs are a way of enforcing that for relevant building elements such information is present and defines how it is encoded. In that sense MVDs are a very relevant component of the pre-checking phase (Eastman et al., 2009) of the model checking workflow in which basic availability of information is asserted. With such assertions in place, the query from listing 4 can be appended with a triple pattern to retrieve and filter on the wall length.

Listing 5. Addition to the SPARQL query to retrieve wall lengths in explicitly stored quantities

```
?rel_prop a ifcowl:IfcRelDefinesByProperties ;
  ifcowl:RelatedObjects_of_IfcRelDefines ?wall ;
  ifcowl:RelatingPropertyDefinition ?prop .

?prop a ifcowl:IfcElementQuantity ;
  ifcowl:Quantities ?quantity .

?quantity a ifcowl:IfcQuantityLength ;
  ifcowl:LengthValue ?length .

?length ifcowl:has_double ?length_value .
```

2.5 Relevance of reasoners

As described in the previous section, MVDs can be seen as a form of contract between the exporting and the importing party of the exchange. On the other hand, it might not always be feasible for exporting parties to be able to comply to all MVDs. Development iterations of exporting software are sometimes lagging behind or views on information are simply not always available in the authoring software. In such cases the importing party may need to infer the information using reasoners. Using reasoners and inference engines, a bottom-up higher-order apprehension of a model can be obtained by inferring new knowledge from explicitly available information and inference rules.

In the case of the rule in Listing 3 and without the explicitly encoded quantity information for wall lengths, which is queried in Listing 5, the length can be obtained by means of a conventional query (Listing 6). This is mainly provided to illustrate the relative infeasibility of this approach.

Listing 6. Addition to the SPARQL query to retrieve wall lengths from geometrical data

```
?wall ifcowl:Representation ?shape .
?shape ifcowl:Representations ?list .
?list ifcowl:isFollowedBy*/ifcowl:hasListContent ?rep .
?rep ifcowl:RepresentationIdentifier ?label .
?label ifcowl:has_string "Axis" .

?rep ifcowl:Items ?rep_item .
?rep_item a ifcowl:IfcPolyline ;
  ifcowl:Points ?point_list .

?point_list ifcowl:hasListContent ?point1 .
?point_list ifcowl:isFollowedBy/ifcowl:hasListContent ?point2 .

?point1 ifcowl:Coordinates_of_IfcCartesianPoint ?p1xy .
?point2 ifcowl:Coordinates_of_IfcCartesianPoint ?p2xy .

?p1xy ifcowl:hasListContent/ifcowl:has_double ?p1x .
?p1xy ifcowl:isFollowedBy/ifcowl:hasListContent/ifcowl:has_double ?p1y .

?p2xy ifcowl:hasListContent/ifcowl:has_double ?p2x .
?p2xy ifcowl:isFollowedBy/ifcowl:hasListContent/ifcowl:has_double ?p2y .

BIND (afn:sqrt((?p2x - ?p1x) * (?p2x - ?p1x) + (?p2y - ?p1y) * (?p2y - ?p1y)) as ?length)

FILTER (?length > 3.0)
```

There is a large amount of complexity involved with querying seemingly trivial information from building models in IFC. This stems from the standard being permissive as it allows geometry to defined in my ways. In addition, complexity is introduced by the objectified relationships that introduce additional indirection to get to relevant information. By introducing shortcuts to such data, the model can be made more idiomatic to common linked data ontologies (Farias et al., 2015). Such shortcuts is also something part of specific-purpose query language for IFC building models, such as presented in (Mazairac and Beetz, 2013).

To summarize, there are three complicating factors illustrated here: 1) the inference of a wall length in this case depends on the availability of an ‘axis’ geometry, which might not be defined, although dictated to exist in textual form 2) for such a representation geometry may be defined in different ways 3) ordered aggregates difficult are difficult to inspect and query in SPARQL as they have to modelled as a tree structure in RDF (Pauwels et al., 2015). Therefore the query in Listing 6 is only able to return a wall length for the simple cases. As such, due to the combinatorial

complexity of these complications, a bottom-up approach to infer new information is more suitable.

3. CONCLUSIONS

In order to come to an efficient automated rule checking workflow, the technical aspects discussed in this paper need studied holistically and not in isolation. By exploiting their complementary nature, rules can be encoded formally, succinctly and expressively.

A crucial foundation for rule checking are the semantics introduced on the schema level: in order to prevent false positives and negatives, elements need to be classified correctly. Currently there are no or very limited provisions in the IFC schema that actually guarantee this formally. As a consequence elements are misclassified, or classified in overly broad or meaningless categories, such as IfcBuildingElementProxy. Elements that are misclassified result either in false negatives, as certain relevant checks are not performed, or in false positives (as illustrated in the case of Figure 1), because irrelevant checks are performed. Both are detrimental to the quality of the rule checking process and therefore to the delivered artefact. For this purpose, a large body of the documentation of the IFC standard, which consists of hundreds of pages, needs to be translated into formal and decidable statements.

Secondly, formal notations of explicitly encoded quantities are necessary in order to guarantee that the available quantities are accurate. IFC is a highly redundant data format: the length of a wall will typically be reflected in many places within the schema, including space boundary geometries, explicitly calculated quantities and different representations for the body and axis of elements. The quality of a data schema can be described by means of a minimal redundancy and maximum reliability (McLeod, 1995). Without formalized connections between the different apprehensions of attributes like the wall length, the amount of redundancy induces an increased risk of errors when parts of a model get updated.

4. REFERENCES

Amor, R., 2015, "Analysis of the Evolving IFC Schema", in: *Proceedings of the 32nd International Conference of CIB W78, Eindhoven, The Netherlands*, p. 39-48.

Eastman, C., Lee, J.-M., Jeong, Y.-S., and Lee, J.-K., 2009, "Automatic rule-based checking of building designs", *Automation in Construction 18(8)*, p. 1011-1033.

Farias, T.M. de, A. Roxin, and C. Nicolle, 2015, "IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties", in: *proceedings of the 32nd CIB W78 Conference on Information Technology in Construction, 2015, Eindhoven, Netherlands*.

Khemiani, L., 2005, "CORENET e-PlanCheck: Singapore 's Automated Code Checking System", AECbytes, Building the Future , 1–8.

Krijnen, T. and J. Beetz, 2016, "An assessment of linked data ontologies for use in the AEC/FM domain", in: *Proceedings of the 33nd International Conference of CIB W78, Brisbane, Australia*.

Krijnen, T. and M. Tamke, , 2015, "Assessing Implicit Knowledge in BIM Models with Machine Learning", in: Modelling Behaviour, Springer International Publishing, p. 397-406.

Lee, Y.C., C.M. Eastman, and W. Solihin, 2016, "An ontology-based approach for developing data exchange requirements and model views of building information modeling", *Advanced Engineering Informatics 30(3)*, p. 354-367.

Mazairac, W. and J. Beetz, 2013, "BIMQL – An open query language for building information models", *Advanced Engineering Informatics 27(4)*, p. 444-456.

McLeod, R.J., 1995, *Management Information Systems: a Study of Computer-Based Information Systems*, 6th Edition, Prentice-Hall, Englewood Cliffs, NJ.

Miller, G.A., 1995, "WordNet: a lexical database for English." *Communications of the ACM 38.11*, p. 39-41.

Navigli, R. and S.P. Ponzetto, 2012, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network", *Artificial Intelligence 193*, p. 217-250

Nawari, O., 2012, Automated Code Checking in BIM Environment, in: *Proceedings of 14th International Conference on Computing in Civil and Building Engineering*

Palos, S., A. Kiviniemi, and J. Kuusisto, 2014, "Future perspectives on product data management in building information modeling", *Construction Innovation 14(1)*, p. 52–68.

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

Pauwels, P., W. Terkaj, T. Krijnen, and J. Beetz, 2015, "Coping with lists in the ifcOWL ontology", in: *Proceedings of the 22nd EG-ICE International Workshop, Eindhoven, The Netherlands*, p. 113-122.

Solihin, W. and C. Eastman, 2015, "Classification of rules for automated BIM rule checking development", *Automation in Construction 53*, p, 69-82.

Solihin, W., C. Eastman, Y.-C. Lee, 2015, "Toward robust and quantifiable automated IFC quality validation", *Advanced Engineering Informatics, 29(3)*, p. 739-756

Terkaj, W. and A. Šojić, 2015, "Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology", *Automation in Construction 57*, p. 188-201

Zhang, C., J. Beetz, and M. Weise, 2014, "Model view checking: automated validation for IFC building models." *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM*.