A Checking Approach for Distributed Building Data

Jeroen Werbrouck[1, 2], Madhumitha Senthilvel[1], Jakob Beetz[1] and Pieter Pauwels[2]

[1] Chair of Design Computation · Schinkelstraße 1 · 52062 Aachen ·
E-Mail: werbrouck@caad.arch.rwth-aachen.de

[2] Dept. of Architecture and Urban Planning · Ghent University, Belgium

Since the early 2000s, the building industry has been steadily embracing the concept of Building Information Modelling (BIM). Currently, the BIM focus lies on file-based collaboration, although with the rise of semantic web technologies, the benefits of web- and data-based collaboration for the Architecture, Engineering and Construction (AEC) industry come within reach. A web-based AEC industry that relies on Linked Data can provide various advantages compared to 'classic' BIM practice, e.g. regarding interdisciplinarity, linking across domains and logical reasoning. In this paper, we investigate Linked Data rule checking mechanisms on decentralised building datasets. The recent Semantic Web standard Shapes Constraint Language (SHACL) is used to check a Linked Data building model that is hosted on multiple data pods. After a short introduction to Linked Building Data and rule checking approaches, a minimal distributed building model will be checked with basic SHACL patterns, generating a report to inform both end users and tools. In this case study, we make use of the Social Linked Data (Solid) ecosystem, a set of conventions and tools for creating decentralised applications.

Keywords: Linked Building Data, decentralisation, rule checking, SHACL, Solid

1 Introduction

The concept of Building Information Modelling/Management (BIM) has been key to the adoption of smart, object-oriented digital technologies in the Architecture, Engineering and Construction (AEC) industry. Nevertheless, a lot of fragmentation is still present, due to the many stakeholders involved in a project, each with different background and using specialised software tools. To improve interoperability, the Industry Foundation Classes[1] (IFC) were founded in 1997, aimed at encompassing the diversity of activities present in the Building Life Cycle (BLC). However, because of the size of the IFC schema, full compatibility with commercial tools is seldom the case, and a lossless information flow between tools is difficult to achieve (Kiviniemi et al., 2005; Shafiq et al., 2018).

A promising candidate for maximising interoperability with integrated data, is the use of the Semantic Web technology stack, instead of relying on document-based building models (Beetz et al., 2005; Pauwels et al., 2017b; Rasmussen et al., 2017). The Semantic Web (Berners-Lee et al., 2001), which can be thought of as an additional layer upon the classic world wide web, enables to link data of various nature on multiple servers in the form of a directed graph. With the use of the Resource Description Framework[2] (RDF) standard for

---

[1] https://technical.buildingsmart.org/standards/ifc

[2] https://www.w3.org/TR/rdf-primer/

establishing Linked Data graphs, a significant format interoperability gain can be achieved, which is why more and more disciplines are making semantic web-compatible domain models, among which also the AEC industry. Furthermore, semantic web technologies allow linking across domains and advanced checking of regulations and rule sets.

RDF allows to set up a decentralised web ecosystem, where interlinked data is stored on different servers, and applications are decoupled from the data. An example of such ecosystem framework is the Social Linked Data (Solid) project (Berners-Lee and Verborgh, 2018; Mansour et al., 2016). Although focused on social data and applications, it offers a set of specifications and tools that can be applied to other disciplines as well. In the case of the AEC industry, the available project data can be stored personally by the different stakeholders, nevertheless forming a coherent model; the basis for a decentralised Common Data Environment (CDE) (Werbrouck et al., 2019). Such distributed Linked Building Data can then be queried and checked with rulesets, which is the core topic of this paper. After a short introduction to the nature of Linked Data, its implementations for the AEC industry and a discussion on rule checking mechanisms, a minimal configuration is set up for retrieving and checking the data with a ruleset based on the Shapes Constraint Language[3] (SHACL) standard (Knublauch and Kontokostas, 2017).

2 Related Work

2.1 Linked Data

In the introduction, the concept of Linked Data and the Semantic Web was introduced, along with its main data model, namely RDF. An RDF graph can be thought of as basic statements that are coupled together into so-called *triples*, consisting of a *subject*, a *predicate* and an *object*. *Subjects* and *objects* form the nodes of the graph, the *predicate* gives meaning to the edge, pointing from subject to object. To give a Linked Data *resource* a context that is globally identifiable over the web, RDF makes use of Uniform Resource Identifiers (URIs)[4]. URIs form a superclass of the more known Uniform Resource Locator (URL): while a URI *identifies* a resource uniquely, a URL also allows to retrieve it via a web page. To improve readability, it is recommended to make use of *prefixes* when referring to a Linked Data resource. Listing 1 illustrates this concept of linking resources with triples and prefixes, formatted with Turtle notation[5]. The ‘inst’ prefix defines project-specific instances, thereby illustrating the difference between specific instances (ABOX) and general classification concepts (TBOX).

Listing 1: RDF data in Turtle format. (*a* refers to *rdf:type*; the prefix *inst* refers to specific instances)

```
@prefix bot: <https://w3id.org/bot#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix inst: <http://www.example.org/myproject#> .

inst:site1        bot:hasBuilding    inst:building1
inst:building1    bot:hasStorey      inst:storey1;
inst:storey1      bot:hasSpace       space1, space2
```

---

[3] https://www.w3.org/TR/shacl/

[4] https://www.w3.org/wiki/URI/

[5] https://www.w3.org/TR/turtle/

Domain models that organise definitions such as taxonomies and relations between nodes are called *ontologies* or *vocabularies*.[6] Ontologies enable to deduce implicit information from the data that is present in a certain graph, a process called *inferencing*. For instance, by looking up the predicate *bot:hasStorey* from List. 1 in the Building Topology Ontology (BOT)[7] (Rasmussen et al., 2017), one can infer that *inst:storey1* is an instance of *bot:Storey*. Certain ontologies, such as RDF Schema[8] (RDFS) and the Web Ontology Language[9] (OWL) serve as general means for setting rules and restrictions for other ontologies.

2.2 Linked Building Data

The idea that a web-based approach to data would offer several advantages to the AEC industry, has been put forth multiple times already (Beetz et al., 2005; Pauwels et al., 2017b; Rasmussen et al., 2017). Specifically, it is expected to be a game changer in terms of interoperability, linking across domains and logical inference and reasoning (Pauwels et al., 2017b). Interoperability is achieved through mappings or implementations of standards into RDF; an important use case is the conversion of the Industry Foundation Classes (IFC) schema into a Linked Data equivalent: ifcOWL[10] (Pauwels and Terkaj, 2016). Furthermore, in the use of the RDF syntax, multiple disciplines can link their knowledge together, in order to achieve a more complete perspective on the available information. Specifically for the AEC industry, this includes domains that are not (sufficiently) integrated in 'classic' BIM environments: heritage data, GIS, Facility Management, circular economy etc. With this in mind, the W3C Linked Building Data Community Group[11] advises the use of small, modular ontologies that are easy to extend and can be coupled with other ontologies in a flexible way. The core LBD ontology is the Building Topology Ontology (BOT), defining the topological structure of a built or unbuilt asset. Other vocabularies can easily be coupled with the BOT core graph of the building (Schneider et al., 2018). This modularity stands in stark contrast with the ifcOWL ontology: because IFC is such an enormous schema, it is easy to lose oversight and difficult to couple other domains in a flexible way.

As mentioned in Section 2.1, RDF relies on one of the fundamental aspects of the web, namely URIs. This is a fundamentally different approach from identification of concepts via GUIDs or local IDs; it allows to distribute data over the web, referring to this data and retrieving it whenever necessary. It means that graph nodes (whether ontology nodes or instance nodes) do not need to be located on the same server, but can be spread around the globe instead. In a building-related context, such distributed systems allow multiple stakeholders to work collaboratively on a building project, while nevertheless remaining the owner of the data they add to the model; a decentralised CDE. As a result, multiple web services that are disconnected from the data itself can be allowed to access the data and retrieve this data from the data stores of different partners. This can be thought of as a Linked Data equivalent of Model Servers such as the BIMserver (Beetz et al., 2010) or the late BLIS/SABLE initiative (BLIS-project, 2002; Kiviniemi et al., 2005).

---

[6] https://www.w3.org/standards/semanticweb/ontology/

[7] https://w3c-lbd-cg.github.io/bot/

[8] https://www.w3.org/TR/rdf-schema/

[9] https://www.w3.org/OWL/

[10] https://github.com/buildingSMART/ifcOWL

[11] https://www.w3.org/community/lbd/

2.3 Rule Checking approaches in AEC

In the current document-based AEC practice, model checking is typically done through mvdXML, a buildingSMART standard (Chipman et al., 2016). The mvdXML standard defines a way to document the Exchange Requirements for an IFC model, accompanied by a method to validate resulting Model View Definitions (MVDs). The *eXtended Process to Product Modeling* (xPPM) (Lee et al., 2013) proposes a method to integrate MVDs with the BuildingSMART Information Delivery Manual (IDM) standard, in order to stimulate a practice of reusing checking templates. A general classification of different rule-checking approaches for the AEC industry is given in (Solihin and Eastman, 2015).

While in the current industry, mvdXML focuses on the validation of document-based IFC models, a lot of work has been done to implement checking mechanisms based on Semantic Web technologies in an AEC context (Pauwels et al., 2017a, 2011; Zhang et al., 2014). This research relates to several domains of checking: apart from model consistency checking, a BIM model can be checked for regulation compliance or project-specific rules. The recent bimSPARQL mechanism (Zhang et al., 2018) streamlines querying information expressed in ifcOWL and reasoning on BIM-related data. It is based on the W3C standard SPIN[12] (SPARQL Inference Notation), the predecessor of SHACL. Going further on this work regarding rule-checking of building models in a Semantic Web environment, one can investigate how such rule checking mechanisms can be applied in a distributed environment.

In the scenario of a decentralised CDE, model checking is slightly more complex than when all information is stored locally, since the checking engine needs to retrieve all information from the different servers. In a traditional object-oriented system, where data is typically stored at a single location, a so-called Closed World Assumption (CWA) is common. In a CWA, it is assumed that all information is present in the system: if information is not present, it is considered false. This stands in contrast with the Open World Assumption (OWA) used in Semantic Web contexts, where new information can be added to the graph at any time and at any web location. Rule checking is more difficult in such open world context, because one can never really 'close the world' and consider a graph 'complete'. This dichotomy between OWA and CWA is also one of the main differences between OWL and SHACL: while OWL restrictions essentially describe inference patterns in an OWA, SHACL covers data validation and constraints and is one of the few Semantic Web technologies that actually involve a CWA (Knublauch, 2017). A combination of the two technologies in different phases of the checking process can thus be a powerful tandem: in a first phase, OWL inference mechanisms infer implicit statements, while in the next phase 'the world is closed' and the model is validated with SHACL rulesets, called *shapes*.

An example SHACL shape is given in Listing 2. This shape imposes that a complete building structure *must* be present, i.e. a bot:Site instance *must* have at least one instance of bot:Building, which in turn needs at least one bot:Storey that contains a minimum of one bot:Space. Also, the bot:Site must link to a geolocation, using the *bot:hasZeroPoint* predicate. In the BOT ontology, this property refers to a wsg84:Point with latitude (*wgs84:lat*) and longitude (*wgs84:long*). The SHACL shape in Listing 2 will be used for the use case discussed in Section 3.

---

[12] https://www.w3.org/Submission/spin-overview/

Listing 2: SHACL shape for building topology description and georeference.

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix b4r: <https://www.bim4ren.org/shapes#> .
@prefix bot: <https://w3id.org/bot#>.

b4r:SiteShape a sh:NodeShape;
   sh:targetClass bot:Site;
   sh:property [
            sh:path    bot:hasBuilding;
            sh:minCount 1;
            sh:message "A Site must have at least one Building";
   ], [
            sh:path    bot:hasZeroPoint;
            sh:minCount 1;
            sh:maxCount 1;
            sh:message "A Site must have exactly one zero Point"; ].

b4r:BuildingShape
   a sh:NodeShape ;
   sh:targetClass bot:Building;
   sh:property [
            sh:path    bot:hasStorey;
            sh:minCount 1;
            sh:message "A Building must have at least one Storey"; ].

b4r:StoreyShape
   a sh:NodeShape;
   sh:targetClass bot:Storey;
   sh:property [
            sh:path    bot:hasSpace;
            sh:minCount 1;
            sh:message "A Storey must have at least one Space"; ].
```

2.4 The Solid Ecosystem

A lot of work has already been done in the Semantic Web field, in general as well as for the building industry. However, a gap seems to persist between available technologies and domain models on the one side, and implementation on the other (Verborgh, 2018). The recent Solid initiative[13] aims at lowering this threshold, providing a set of specifications for developing Social Linked Data applications along with an infrastructure that preconfigures these specifications. In this way, a developer can focus on the functionality of her app instead of spending precious time at, for instance, Linked Data authentication and authorisation.

As indicated in Section 2.2, disconnecting applications from the data they use allows to configure a network of tools that can connect to the distributed data stores (*pods* in Solid terminology). This concept of decentralisation and personal data pods lies at the core of Solid. Although its primary use case lies in Social Web applications, Solid also lends itself quite well for the building industry, considering the many stakeholders and tasks present in the BLC. Thus, in a similar way, modular chains of Solid-compatible *BIM bots* (van Berlo, n.d.)

---

[13] https://inrupt.com/solid

can be established for specific activities in the Building Life Cycle, in a flexible and automated approach to counter (non-)typical challenges in building projects.

In the next section, the above topics of Linked Building Data, validation and decentralisation will be combined in a minimal use case for checking distributed models.

3 Case Study: checking distributed building models

3.1 Set up

As a case study, we configure a React app that complies with the Solid specs, in order to deal with distributed datasets from multiple stakeholders. The basis for the app is generated with the Solid React Generator. It is a part of the ConSolid app, which aims to test the use of Solid for construction purposes (Werbrouck et al., 2019).

At the moment of writing, this app allows to configure a building project with the use of BOT, to set different stakeholders with the use of their WebID (a method to identify people over the web using URLs) and to link the project documentation from different stakeholders (unstable). The checking functionality discussed in this section uses this data to check the model. Currently, this checking procedure takes the following workflow:

- Look up the stakeholders of the project;
- Fetch the project data (after authorisation) and combine it into a single graph;
- Read the SHACL shape URL from the user input;
- Check the building graph against the SHACL shape;
- Generate a report and present it to the user.

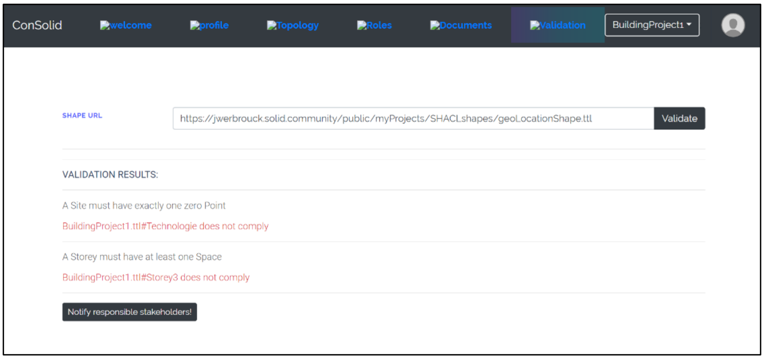
3.2 Validation

For this case study, we configure one pod that contains a BOT representation of the building topology and a graph that contains the stakeholders and their role in the project, and another pod that contains the geolocation of the project, linked to the bot:Site instance in the other pod. This (minimal) distributed graph will be validated against the SHACL shape in Listing 2, which demands a consistent topology (i.e. site, building, storeys and spaces) and a geolocation that is linked to the instance of bot:Site.

For this example, we consider a scenario where one bot:Storey instance does not relate to any instance of bot:Space, and where the surveyor has not set the geolocation yet.

When the manager authenticates and goes to the ‘Validation’ tab, a URL to a SHACL shape can be set. For a flexible reuse of shapes, it is recommended to make it available to anyone. In this case, the shape is located at a public location in the manager pod, but it could as well be stored on any other server (e.g. Github). The resulting report is depicted in Fig. 5.

Figure 5 also indicates that a certain constraint (*sh:NodeShape*) can be linked to specific stakeholders, so they can be notified if the constraint is not satisfied. This may be a default mapping or can be overridden in the project. In this example, the first error is the responsibility of the project surveyor, the second one of the architect. After these stakeholders enriched the graphs with the necessary data, a pass report is generated (Fig. 6).



SHAPE URL https://jwerbrouck.solid.community/public/myProjects/SHACLshapes/geoLocationShape.ttl

VALIDATION RESULTS:

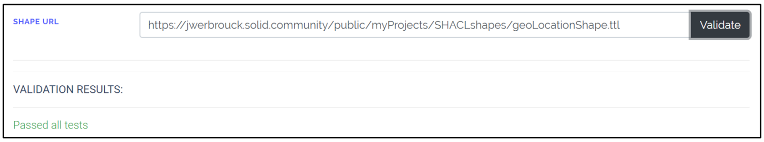
A Site must have exactly one zero Point

BuildingProject1.ttl#Technologie does not comply

A Storey must have at least one Space

BuildingProject1.ttl#Storey3 does not comply

Fig. 5: Validation of the model (“BuildingProject1”) against a given SHACL shape - FAIL



SHAPE URL https://jwerbrouck.solid.community/public/myProjects/SHACLshapes/geoLocationShape.ttl

VALIDATION RESULTS:

Passed all tests

Fig. 6: Validation of the model against a given SHACL shape - PASS

In this section, a very basic scenario was sketched in which a distributed building model, with information owned by diverse stakeholders, can be checked against a ruleset that is online available. When stakeholders can be identified and their responsibilities formalised, the web authentication standards used in Solid can provide a way to grant fine-grained access of project data to certain people involved in the project, retrieve this data and perform different checks on the model.

4 Conclusion and Future work

This paper demonstrated the use of the W3C standard SHACL applied to distributed Linked Building Data for building model checking. After a brief introduction to Linked Data, in general and applied to the AEC industry, a possible approach of using SHACL with the Solid framework was suggested. Using existing infrastructure and standards, setting up such validator is a quite straightforward process. However, the following notes can be made and need to be checked in future research:

- What are the limits of SHACL validation? Is it possible to configure shapes that check building regulations, project ambitions or geometric constraints (e.g. accessibility)? Can we make use of BimSPARQL?
- How to integrate efficient OWL reasoning, prior to SHACL validation?
- Does the system still perform well enough with large, real-world datasets?
- Is real-time cooperation on such models possible?
- Is there a way to present such interface to non-Linked Data experts, i.e. the majority of AEC professionals.

The checking application now forms part of the ‘ConSolid’ testing app in development. In a more advanced scenario, the different modules of this app are not as interlinked as in this proof-of-concept. Rather, they are set up as middleware APIs that can be used for multiple front-end applications and be written in the developer’s preferred language. This will result in a flexible (re)use of modules and enables the configuration of different tool chains, each one dedicated to a particular use case. Validation technologies may be essential in such chains, to validate if the input data (which comes from unknown web locations) is fit to use within the API method. This scenario should be the subject of future research as well.

This paper featured only a very lightweight model and SHACL shape, so upscaling this to a more realistic building context is necessary. However, we illustrated some basic features and some opportunities for further research can be identified: the combination of decentralisation, distributed building data, modular ontologies and modular applications could be the basis for an ecosystem for managing the diverse facets of the BLC, with Linked Building Data.

5 Acknowledgements

The authors would like to acknowledge the support by both the BIM4REN project, which has received funding from the European Union’s Horizon 2020 Research and Innovation Program under grant agreement No 820773, and the UGent Special Research Fund (BOF).

References

Beetz, J., van Berlo, L., de Laat, R., van den Helm, P., 2010. BIMserver. org–An open source IFC model server, in: Proceedings of the CIP W78 Conference. p. 8.

Beetz, J., Van Leeuwen, J.P., De Vries, B., 2005. An Ontology Web Language Notation of the Industry Foundation Classes, in: Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction. Technical University of Dresden, p. 670.

Berners-Lee, T., Hendler, J., Lassila, O., 2001. The Semantic Web. Scientific american 284, 28–37.

Berners-Lee, T., Verborgh, R., 2018. Solid: Linked Data for personal data management.

BLIS-project, 2002. BLIS Building Lifecycle Interoperable Software. (http://www.blis-project.org/index2.html)

Chipman, T., Liebich, T., Weise, M., 2016. mvdXML - specification of a standardized format to define and exchange Model View Definitions with Exchange Requirements and Validation Rules.

Kiviniemi, A., Fischer, M., Bazjanac, V., 2005. Integration of multiple product models: Ifc model servers as a potential solution, in: Proc. of the 22nd CIB-W78 Conference on Information Technology in Construction.

Knublauch, H., 2017. SHACL and OWL Compared. (https://spinrdf.org/shacl-and-owl.html)

Knublauch, H., Kontokostas, D., 2017. Shapes Constraint Language (SHACL). (https://www.w3.org/TR/shacl/)

Lee, G., Park, Y.H., Ham, S., 2013. Extended Process to Product Modeling (xPPM) for integrated and seamless IDM and MVD development. Advanced Engineering Informatics 27, 636–651.

Mansour, E., Sambra, A.V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., Berners-Lee, T., 2016. A Demonstration of the Solid Platform for Social Web Applications, in: Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 223–226.

Pauwels, P., De Farias, T.M., Zhang, C., Roxin, A., Beetz, J., De Roo, J., Nicolle, C., 2017a. A performance benchmark over semantic rule checking approaches in construction industry. Advanced Engineering Informatics 33, 68–88.

Pauwels, P., Terkaj, W., 2016. EXPRESS to OWL for Construction Industry: Towards a recommendable and usable ifcOWL Ontology. Automation in Construction 63, 100–133.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., Van Campenhout, J., 2011. A semantic rule checking environment for building performance checking. Automation in Construction 20, 506–518.

Pauwels, P., Zhang, S., Lee, Y.-C., 2017b. Semantic Web Technologies in AEC Industry: A Literature Overview. Automation in Construction 73, 145–165.

Rasmussen, M., Pauwels, P., Lefrançois, M., Schneider, G.F., Hviid, C., Karlshøj, J., 2017. Recent Changes in the Building Topology Ontology, in: LDAC2017-5th Linked Data in Architecture and Construction Workshop.

Schneider, G.F., Rasmussen, M.H., Bonsma, P., Oraskari, J., Pauwels, P., 2018. Linked building data for modular building information modelling of a smart home, in: 11th European Conference on Product and Process Modelling. CRC Press, pp. 407–414.

Shafiq, M.T., Matthews, J., Lockley, S., Love, P.E., 2018. Model server enabled management of collaborative changes in building information models. Frontiers of Engineering Management 5, 298–306.

Solihin, W., Eastman, C., 2015. Classification of rules for automated BIM rule checking development. Automation in construction 53, 69–82.

van Berlo, L., n.d. bimbots.org - Automate your work with BIM Bots. (http://bimbots.org)

Verborgh, R., 2018. Designing a Linked Data Developer Experience. (https://ruben.verborgh.org/blog/2018/12/28/designing-a-linked-data-developer-experience/)

Werbrouck, J., Pauwels, P., Beetz, J., van Berlo, L., 2019. Towards a Decentralised Common Data Environment using Linked Building Data and the Solid Ecosystem. Presented at the Proceedings of the 36th CIB W78 Conference on Information Technology in Construction.

Zhang, C., Beetz, J., de Vries, B., 2018. BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. Semantic Web 1–27.

Zhang, C., Beetz, J., de Vries, B., 2014. An ontological approach for semantic validation of IFC models, in: Proceedings of the 21st International Workshop on Intelligent Computing in Engineering (EG-ICE).