Attendees:

- Rui de Klerk (FAUL)
- Kris McGlinn (TCD-ADAPT)
- Sander Stolk (Semmtech)
- Gonçal Costa (LaSalle University)
- Håkon Reisvang (i4 technology Norway)
- Joel Bender (Cornell University)
- Mathias Bonduel (KU Leuven)
- Seppo Törmä (VisuaLynk)
- María Poveda-Villalón (UPM, Spain)
- Jan Voskuil
- Calin Boje
- Katja Breitenfelder
- Jason Loh (BRE, UK)
- Mads Holten Rasmussen (Niras, DK)
- Aaron Costin (University of Florida, USA)
- Jeroen Werbrouck (UGent)
- Andrew Malcolm (UGent)
- Hervé Pruvost (Fraunhofer IIS EAS)

Date and time

- 26/11/2019
- 15:00 WET/ 16:00 CET

Agenda

1. Introductions for new attendees (open)
2. Use case gathering (initial focus geometry) (Kris, Mathias, Anna)
3. ifcOWL Validation using SHACL (Sander Stolk)
4. Other items (open)

Minutes

1. Introductions for new attendees (open)
2. Use case gathering (geometry) (Kris, Mathias, Anna)
    - Google Form (for editing)
        i. https://docs.google.com/forms/d/1EaKYOF6HOL6A1SIJCtxjvBEqPX0bBZwCSFd7zuuBSqo/edit
    - Google Form (for sharing to fill in)
        i. https://docs.google.com/forms/d/e/1FAIpQLScMvRq9UBkqi2QhBpmkO6AzTe7yK_n5O6MmDY-SiJ9wLHfQeA/viewform?usp=sf_link

    - Sample
        i. https://docs.google.com/document/d/1StDsrIyVvwyE6KdyUlDyS4gM52Yot7QAAVcui705MXs/edit#
3. ifcOWL Validation using SHACL (Sander Stolk)

Presentation will cover differences between OWL inferences and SHACL rules

Slides can be found here:

https://drive.google.com/file/d/1qPonm9LsAQzsIXMZHrcHdKdacl344HTZ/view?usp=sharing

Brief summary:

- Inference w. RDFS + OWL
    - Taxonomical inference: rdfs:subClassOf
    - Property inference: rdfs:domain, rdfs:range
    - Ontological inference: owl:Restriction, owl:allValuesFrom, owl:minCardinality
    - Summary: RDFS & OWL add information that can be inferred, even if not directly asserted
- What about for validation, i.e. proving that something is correct or approved (complies with a schema for example
    - Use of validation shape - i.e. a set of constraints that indicate what patterns asserted data must follow
- SHACL validation: property, path, class
    - Property - leads to a resource or property shape that leads to a certain class, e.g. MotherVehicle has engine Engine
    - VehicleX has engine <untyped class> - SHACL will complain if the object of a triple is not asserted correctly.
    - So, if you have a SHACL constraint that says any instance of a car must have four wheels recorded/asserted, and these wheels are not asserted somewhere, SHACL will throw an error.
    - Typical use case, handovers from a contractor to client, datasets should already contain all necessary information, so closed world assumption applies.
- How can we validate an ifcOWL model so that it adheres to the IFC schema? For example, if we wish to convert ifcOWL back to STEP.
    - SHACL offers standardized validation tools
    - If we look at the EXPRESStoOWL converter.
    - The shaclvalidate tool returns a report on which rules are broken.
    - So when expressing ifcOWL, we can return what aspects of the ifcOWL model may be missing when generating ifcOWL models.
- Some restrictions which some decisions need to be made on:
    - ifcowl:lengthValue_IfcQuantityLength -> a) express:REAL b)IfcLengthMeasure c) express:INTEGER

    - In SHACL we should state that a length value should always go to a REAL or its subclasses, which would indicate that case c) would be in violation

Discussion:

Mathias: Should we express SHACL rules for BOT and ontologies under development as part of the LBD group.

Sander: You could indicate what data is required to support a specific API or application. This could be defined as a shape. In the case of ifcOWL we can support going to and from ifcOWL and back to STEP. For BOT, if we want to support applications we can define SHACL rules and they can test to make sure that the data generated with will work with the the app.

Jan: It is a good idea to have shapes alongside the classes and properties. In our experience, when working with ontologies, we tend to define resources as a class and also a node shape (property shapes). This specify all the things we need to know about what properties they should have. This could be added as an extra layer. May be more important that OWL, as validation is often more important than inference.

Walter: In the past I did some work on validation (2015 paper validation of ifcOWL). We used a tool called integrity constraint validator (pellet) which is now included in stardog.

- Have you compared the performance of these validators with SHACL?
- In EXPRESS to OWL conversion, SHACL could be used for the parts of the express schema outside ot ifcOWL (express where rules). Could be good to include into SHACL rules.

Sander: I have not compared the processing speed, so cannot compare. These comparisons may exist, for example some JS implementations. SHACL should be relatively quick, as this is in essence pattern checking, no inference is required.

Using SHACL to support other express schema aspects sounds like an interesting research question.

Jan: These tools are not standard compliant, and can behave in different ways. SHACL can express more than OWL restrictions, e.g. a value must be greater than another.

Mads: Advanced SHACL?

Jan: The advanced SHACL gives you all the things which SPIN could do, a way to work with/process data using SPARQL. When it comes to manipulating, cleansing over data then SHACL rules is a convenient tool. There are commercial implementations, and open source.

Mads: Anything for JS?

Jan: I can look into that.

4. Other items (open)

None

Previous minutes

https://docs.google.com/document/d/1tFYXICwVh37tU-E97rTsBbPQA9zPCITqjCyDoUmT8rk/edit#heading=h.k621biyoh9tu

Next Call

- [Tentative] early December 2019 ->
    - WLS, north american LDAC by Aaron Costin,
    - actions in cen tc 422 (Peter Kompolschek)
    - Revision and collection of use cases