Building Data on the Web Community Group

Subgroup Building Properties (First Draft)

Contents

Introduction

The following document is the working document of the Subgroup Building **Properties** of the W3C Building Data on the Web Community Group. For more details on the community group please visit the charter and the community group home page.

Members

Current members of the sub-group are:

- Mathias Bonduel
- Michel Böhms
- Claudio Benghi
- Ana Roxin
- Mads Holten Rasmussen

- Georg Ferdinand Schneider
- ...

The subgroup as well as the community group are open for all interested participants. Please join here (for the community group) or send an email to the current sub-group members (see list above).

Scope

The Building Properties subgroup focuses on the representation of building-related properties using semantic web technologies. A building-related property is defined here as a property of a physical or virtual building element (bot:Element), 3D spatial zone (bot:Zone), an interface between an element and a zone (bot:Interface) and a product (product:product).

Not In Scope

The following information areas are explicitly not within the scope of the subgroup:

- Geometric data (properties derived from <some> geometry are included, but the geometry itself not)
- Properties of things that are not present in a building (focus area: buildings)
- Commercial and sales properties that are needed for selling an actual product (cfr. GoodRelations - schema.org)

Current Action Items

1. Use Case Development
2. Ontology structure - patterns
3. Ontology content

Working Draft

Representation of Properties: the PROPS ontology

For the moment, no ontology is published on the Github page, as the group first has to decide on the ontology structure (patterns) and the content. A basis ontology will be published afterwards.

The name of the ontology would better be not affiliated directly with IFC as the scope of the ontology is wider than only IFC properties. The initial name (PSET ontology) is for this reason discarded.

The main concept is explained now in <gihtub repo>, but it should have a written version with illustrations here as well.

Each of these ontologies is available within the same namespace: <PROPS namespace>

An **overview of the current approaches** to PROPS was presented in the plenary call of the LBD Community Group on the 12th of March:

- Presentation
- Minutes

Ontology for properties management (OPM)

This ontology is proposed for managing changing property values over a project or building lifecycle. The OPM approach in fact extends the existing SEAS ontology Evaluation concept.

More information to be added
https://w3id.org/opm

Relations with outside ontologies

The BOT topology ontology

The PRODUCT ontology

The schema.org (swallowed GoodRelations) ontology

SEAS ontology

Units ontologies: QUDT, OM, O&M, CDT

Examples

The following list of examples was presented during the plenary LBD Community Group call of the 12th of March. The sparql-visualizer application, developed by Mads, was used to demonstrate current approaches to building-related properties and work-in-progress of investigated use cases. The open-source application is available online, while documentation (incl. 2 tutorial videos and a quickstart) can be found on the Github page . No software has to be installed; everything runs directly in your browser.

Current approaches

- ifcOWL

- IfcWoD and simpleBIM
- BIMSO-BIMDO
- (sensors)
- (schema.org)
- ...

Investigated WIP

- IFCtoLBD converter
- Building requirements + alignment with schema.org
- …

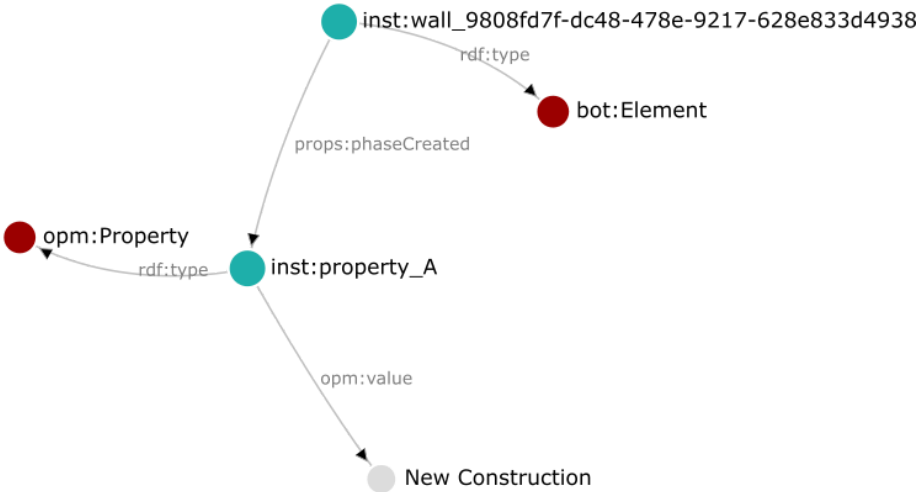
Everyone is invited to provide samples of their own, preferably by using the sparql-visualizer app as a medium to quickly communicate ideas. Share these samples in the issues of the PROPS github repository. The links to these samples can also be used when making a use case or requirement submission in the following document: https://w3c-lbd-cg.github.io/lbd/UseCasesAndRequirements/#UseCases

Open issues after plenary call of 12/03/2018

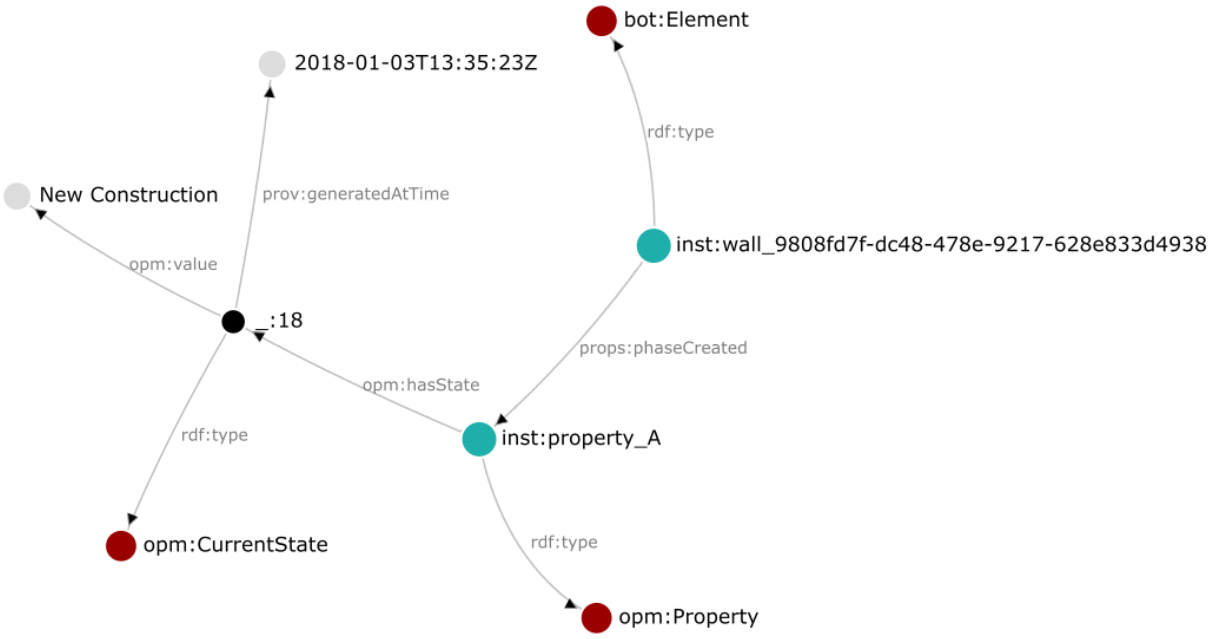
Ontology **structure** (patterns)

- How do we deal with materials? (in scope of PROPS ontology or other ontology?)
    - Inspiration in BIMSO-BIMDO approach: materials as instances that have properties of their own
- Competency questions (more can be added or refined):
    - Query execution time
    - Easy/intuitive to discover properties
    - **Reasoning => what do we want to be inferred?**
    - **Alignment with other existing ontologies (e.g. schema.org)**
    - **PROPS ontology should be easy to maintain and extend**
    - Extra information about a property (description, label, validity, etc.)
    - Grouping props (e.g. IFC psets)
    - Versioning of props
    - Units for props
    - literals: data typing (integer, float, boolean, strings, etc.) + language tags (strings)
    - **Complex props: depending on other props via a math function + table of values**
- Different levels of complexity: one ⇔ multiple
    - **Level 1**: 1 step/relation = 1 object/datatype property

    - **Level 2**: 2 steps/relations = 1 object property + 1 object/datatype property

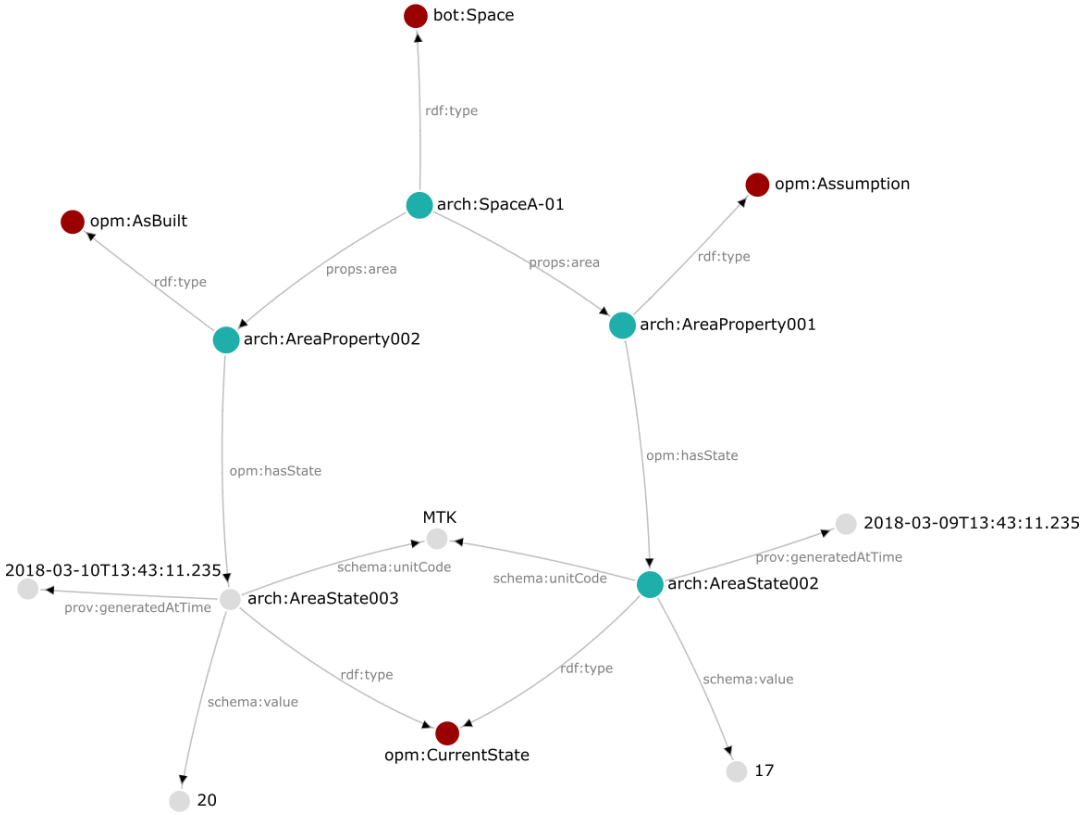


- **Level 3**: 3 steps/relations = 2 object properties + 1 object/datatype property

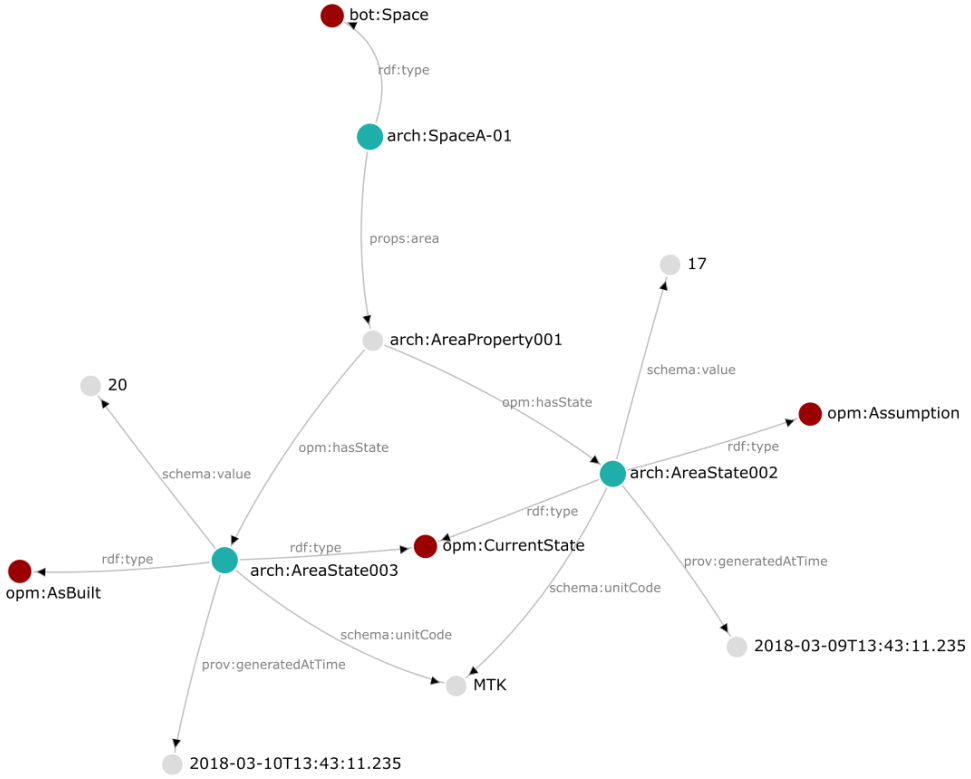


- Multiple levels of complexity seems good to leave the user with the flexibility to use whatever solution fits them
  - This implicates that standardized conversions should be proposed
    - SPARQL INSERT/DELETE and CONSTRUCT
    - (Reasoning / rules)
- All lvls of complexity in one ontology?
  - as the same property cannot be defined as a object property and later on in the ontology as a datatype property, it would be good to distinguish between them. Mads proposed to add a suffix "_simple" to the end of the property name, to indicate that it's a datatype property of the original object property (e.g. props:area and props:area_simple)
    - Should definitions, labels of properties by duplicated between the datatype and object properties? If they should be reused, how do we do it correctly?
- How to define units in the case of level 1?
  - Fixed for every instance of a certain property
    - Non-formal: As human-readable text in the PROPS ontology (only one unit can be defined)
    - Formal: defined in the PROPS ontology by connecting the datatype property
  - Alternative: using CDT (custom datatypes)
    - Now in draft version => only usable when it becomes a standard and software/databases have implemented it?
    - What happens with the xsd datatypes when using CDT? (e.g. xsd:float, xsd:integer, etc.) => every number becomes a xsd:decimal?

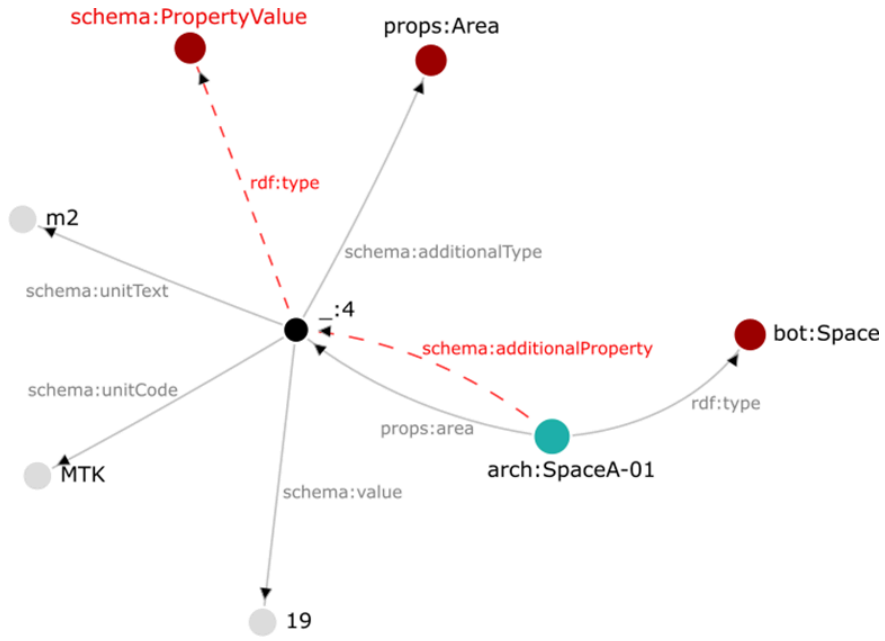
- Units and additional information about the property (e.g. defining a property as an assumption, an as-build value, an as-designed value) should be defined where in the case of level 2 and level 3? On the property instance URI ⇔ the state URI
    - For compatibility reason between level 2 and 3, it seems better to do it on property instance level



    - If only the most complex level (lvl 3) would exist, it would make sense to define the extra information (stating it's an assumption, an as-designed, as-built value) on the state level, in order to avoid duplication of information (compare above graph with the one below)
        - If you do it on the property instance, it would add a lot of different property branches from the building element (one branch for properties that are assumptions, one for as-designed properties, etc.)



- Open issues regarding alignment with schema.org (GoodRelations)
    - Can every level of complexity be aligned with schema.org? Currently an approach regarding level 2 was proposed in the plenary LBD call presentation, slide 21 (12th of March)



    - schema:unitCode has text and URL as range => normally also possible to use it in combination with QUDT / OM / …. See example where QUDT is used

    - There is duplication of information if both props:area (object property, more intuitive to query) and props:Area (owl:Class, necessary for alignment with schema.org) are needed
        - Possible to set the range of props:area to be props:Area class
    - schema:PropertyValue and schema:additionalProperty are necessary to allign with schema.org
        - Possible to set props:area as a subproperty of schema:additionalProperty
        - Possible to set props:Area as a subclass of schema:PropertyValue
    - Schema.org seems to propose a typical key-value pair (using schema:name), when using schema:additionalProperty (see graph below, taken from [this discussion] on Github repository of schem.org). What happens if states (lvl of complexity 3) is used: opm:hasState ⇔ schema:value?

Ontology **content**:

- Start from scraping IFC schema to generate a first PROPS ontology
    - Which schema (IFC2x3 - IFC4)?
    - Some properties (IFC attributes) are not defined in property sets (Psets): they should be differentiated from the ‘regular’ properties in psets. Proposal: add a suffix to the name of the property (e.g. props:name_attribute for the attribute and props:name for the property in a certain pset)
- How can names and definitions of properties from standards (ISO - CEN - national - ...) be integrated?
- Find alignments with other ontologies defining PROPS => we have to watch out with different meanings of similar looking properties in these other ontologies
    - ~~**QUDT**~~ => properties URI exist (e.g. qudt:Volume) but there is no information attached to it (definition, labels, domain, range, etc.). This should be followed up, as the QUDT v2 is under active development

    - **OM** => the properties are defined as owl:Classes
    - **Schema.org** => at first sight only very generic properties (with e-commerce specific meaning, e.g. schema:area is a property of a broadcasting system)
    - **Wikidata** => issue with volatile nature of this dataset (everyone can change it)
- Ontology content should be extendible as it's not possible to capture all possible properties
    - Describe clear methods on how to extend the ontology
    - What properties should be in the published ontology? => list with basis properties where users can extend from (e.g. via subproperties)?

In general: more use cases and existing approaches towards the use of PROPS are needed, such as:

- PROPS in the case of sensors: how to differentiate between properties of the sensor object and the measurements of the sensor (SSN/SEAS/SAREF/…)
- The use of PROPS in the field of e-commerce (e.g. schema.org - GoodRelations approach)
- The use of PROPS in fields that are related to the construction domain, e.g. mechanics, infrastructure engineering, etc.