OPM

Ontology for Property Management

Mads Holten Rasmussen1,2 | M Bonduel3 | M Lefrançois4 | C A Hviid1 | J Karlshøj4

NIRAS1 | DTU2 | KU Leuven3 | MINES Saint-Étienne4

June 19th 2018
Background
Building design

Planning → Design → Construction
Building design in reality

Planning

Design

Construction
Still <Window-325> ??
<Window-325>

width: 1200 mm
panes: 2

<Window-325>

width: 1800 mm
panes: 3
How to allow changes while still being able to maintain history?
<Window-325> "1800 mm"
<table>
<thead>
<tr>
<th></th>
<th>Units</th>
<th>Metadata</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>✔</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>L2</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
</tr>
<tr>
<td>L3</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
03
Using OPM
First state
Second state

```
<state1>
  | prov:generatedAtTime
  | prov:CurrentPropertyState
  | rdf:type
  | opm:CurrentPropertyState
  | schema:value
  | seas:Evaluation
  | prov:Entity, opm:PropertyState
  | "some value"
  |
<prop>
  | prov:generatedAtTime
  | prov:CurrentPropertyState
  | rdf:type
  | opm:CurrentPropertyState
  | schema:value
  | seas:Evaluation
  | prov:Entity, opm:PropertyState
  | "new value"
  |
<state2>
  | prov:generatedAtTime
  | rdf:type
  | opm:CurrentPropertyState
  | schema:value
  | seas:Evaluation
  | prov:Entity, opm:PropertyState
  | "2018-03-23T13:00:00Z"^^xsd:dateTime
  |
```

new state

previous state
Second state

```
"2018-03-23T13:00:00Z"^^xsd:dateTime
new state
previous state
```

```
DELETE {
?previousState a opm:CurrentPropertyState
}
INSERT {
?propURI opm:hasPropertyState ?stateURI .
?stateURI a opm:CurrentPropertyState ;
prov:generatedAtTime ?now ;
schema:value ?val .
}
WHERE {
BIND(<prop> as ?propURI) # define URI of Property
BIND("new value" as ?val) # define new value
BIND(<wall_A#state2> as ?stateURI) # define URI for State
BIND(NOW() as ?now) # get current time stamp

?propURI opm:hasPropertyState ?previousState .
?previousState a opm:CurrentPropertyState ;
schema:value ?currentVal . # get value of current state
FILTER(?val != ?currentVal) # don’t update if equal to latest state
```
Third state (delete)
Third state (delete)

---

Previous state

---

Deleted state

---

Current state

---

Third state (delete)
Fourth state (restore)
Fourth state (restore)

DELETE {
    ?previousState a opm:CurrentPropertyState
}
INSERT {
    ?propURI opm:hasPropertyState ?stateURI .
    ?stateURI a opm:CurrentPropertyState ;
    prov:generatedAtTime ?now ;
    ?key ?val .
}
WHERE {
    BIND(<wall_A#prop> as ?propURI)  # define URI of Property
    BIND(<wall_A#state4> as ?stateURI)  # define URI of new State
    BIND(NOW() as ?now)                # get current time stamp
    # get time stamp of most recent property state that was not deleted
    { SELECT ?propURI (MAX(?time) AS ?t)
    WHERE {
        ?propURI opm:hasPropertyState ?s .
        ?s schema:value ?lastVal ;
        prov:generatedAtTime ?time .
        MINUS { ?s a opm:Deleted }
    } GROUP BY ?propURI }
    # get key-value pairs of latest state that is not deleted
    ?propURI opm:hasPropertyState [  
        prov:generatedAtTime ?t ;
        ?key ?val ]
    FILTER(?key != prov:generatedAtTime)  # filter out time stamps
    # get previous state
    ?propURI opm:hasPropertyState ?previousState .
    ?previousState a opm:CurrentPropertyState .
}
L1, L2, L3 in same graph
Proof of Concept
Architect’s dataset
ICE engineer’s dataset
HVAC engineer’s dataset

bot:Space
- bot:Zone
  - props:area
    - prov:generatedAtTime: "2018-03-15T09:17:38"
    - cdt:area: "20 m²"
  - opm:hasPropertyState
  - rdf:type
    - bot:Element
      - fso:Heater
        - rdf:type
          - bot:Element
            - fso:heatedBy
            - props:heatOutput
              - prov:generatedAtTime: "2018-03-21T07:15:15"
              - cdt:power: "440 W"
        - rdf:type
          - bot:containsElement
            - fso:heaterBy
              - opm:CurrentPropertyState
  - prov:wasDerivedFrom
    - opm:CurrentPropertyState
      - schema:value
        - rdf:type
          - prov:generatedAtTime: "2018-03-18T11:12:25"
          - cdt:power: "400 W"
## Implementations

<table>
<thead>
<tr>
<th>Tool</th>
<th>Author</th>
<th>Repository Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revit BOT Exporter</td>
<td>Mads Holten Rasmussen</td>
<td><a href="https://github.com/MadsHolten/revit-bot-exporter">https://github.com/MadsHolten/revit-bot-exporter</a></td>
</tr>
<tr>
<td>IFC-LBD</td>
<td>Jyrki Oraskari, Mathias Bonduel</td>
<td><a href="https://github.com/jyrkioraskari/IFCtoLBD">https://github.com/jyrkioraskari/IFCtoLBD</a></td>
</tr>
<tr>
<td>Query Generator</td>
<td>Mads Holten Rasmussen</td>
<td><a href="https://github.com/MadsHolten/opm-qg">https://github.com/MadsHolten/opm-qg</a></td>
</tr>
</tbody>
</table>
The numbers

<table>
<thead>
<tr>
<th></th>
<th>This work</th>
<th>OPM Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Object properties</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Datatype properties</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Logical axioms</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>DL Expressivity</td>
<td>$\text{ALCH}$</td>
<td>$\text{ALCH(D)}$</td>
</tr>
</tbody>
</table>
Conclusion

- there is a demand for tracking history during design of buildings
- possible to store every state of a property
- possible to relate a derived property to the very state of the argument it is based on
- future work on derived properties
  - consequence analysis
  - who is influenced?