Model View Definitions (MVDs) and Linked Building Data

2016 buildingSMART International Standards Summit

Jeju, Korea

Technical Room

Pieter Pauwels, Ghent University

28 September 2016

11:00 – 13:00

LDWG

buildingSMART International home of openBIM

1. MVD background
2. The motivation
3. Publishing (static) versus querying (dynamic)
4. White paper

Outline

MVDXML structure

```
<?xml version="1.0"?>
<mvdXML>
   <Templates>
         <ConceptTemplate >
               <Definitions />
               <Rules />
               <Subtemplates />
         </ConceptTemplate >
   </Templates>
   < Views >
      <ModelView />
      <Roots>
            <ConceptRoot>
                  <Concept>
                        <Template />
                        <Requirements />
                        <Rules />
                  </Concept>
            </ConceptRoot>
      </Roots>
   </Views>
</mvdXML>
```

MVDXML structure

```
<?xml version="1.0"?>
<mvdXML>
   <Templates>
        <ConceptTemplate >
             <Definitions />
             <Rules />
             <Subtemplates />
        </ConceptTemplate >
   </Templates>
   < Views >
      <ModelView />
      <Roots>
          <ConceptRoot>
              <Concept>
                   <Template />
                   <Requirements />
                   <Rules />
              </Concept>
          </ConceptRoot>
      </Roots>
   </Views>
</mvdXML>
```

CONCEPTTEMPLATE
This lists HOW content should be included in the subset

LINK
EACH concept linked to ONE ConceptTemplate

CONCEPT
This lists WHAT content should be included in the subset

The template in an MVDXML

```
<ConceptTemplate>
<Definitions />
<SubTemplates>
    <ConceptTemplate uuid="f7cb0a08-fee7-4cc0-85fe-43f6a6a0bce1" name="Identity" status="" applicableSchema="IFC4" applicableEntity="IfcRoot">
        <Definitions />
        <Rules>
            <AttributeRule AttributeName="GlobalId" Cardinality="_asSchema">
                <EntityRules>
                    <EntityRule EntityName="IfcGloballyUniqueId" Cardinality="_asSchema" />
                </EntityRules>
            </AttributeRule>
            <AttributeRule AttributeName="Name" Cardinality="_asSchema">
                <EntityRules>
                    <EntityRule EntityName="IfcLabel" Cardinality="_asSchema" />
                </EntityRules>
            </AttributeRule>
            <AttributeRule AttributeName="Description" Cardinality="_asSchema">
                <EntityRules>
                    <EntityRule EntityName="IfcText" Cardinality="_asSchema" />
                </EntityRules>
            </AttributeRule>
        </Rules>
    </ConceptTemplate>
</SubTemplates>
</ConceptTemplate>
```

Data Exchange Usability

```
@prefix :      <http://www.buildingsmart-tech.org/ifcOWL#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix mvd:   <http://linkedbuildingdata.net/ifc/rules/20150630_144046/> .
@prefix dce:   <http://purl.org/dc/elements/1.1/> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix inst:  <http://linkedbuildingdata.net/ifc/instances20150609_015154#> .
@prefix log:   <http://www.w3.org/2000/10/swap/log#> .

inst:IfcLabel_273230
                a  mvd:IfcLabel ;
                mvd:has_string  "Ghent" .

inst:IfcLabel_273233
                a  mvd:IfcLabel ;
                mvd:has_string  "Belgium" .

inst:IfcBuilding_36
                a       mvd:IfcBuilding ;
                mvd:BuildingAddress  inst:IfcPostalAddress_35 .

inst:IfcPostalAddress_35
                a           mvd:IfcPostalAddress ;
                mvd:Country  inst:IfcLabel_273233 ;
                mvd:Region   inst:IfcLabel_273231 ;
                mvd:Town     inst:IfcLabel_273230 .

inst:IfcLabel_273231
                a  mvd:IfcLabel ;
                mvd:has_string  "Flanders" .
```

MVD building location - IfcBuilding

1. MVD background
2. The motivation
3. Publishing (static) versus querying (dynamic)
4. White paper

Outline

GAP

IFC-SPF

MVD subset

Simple Query Access

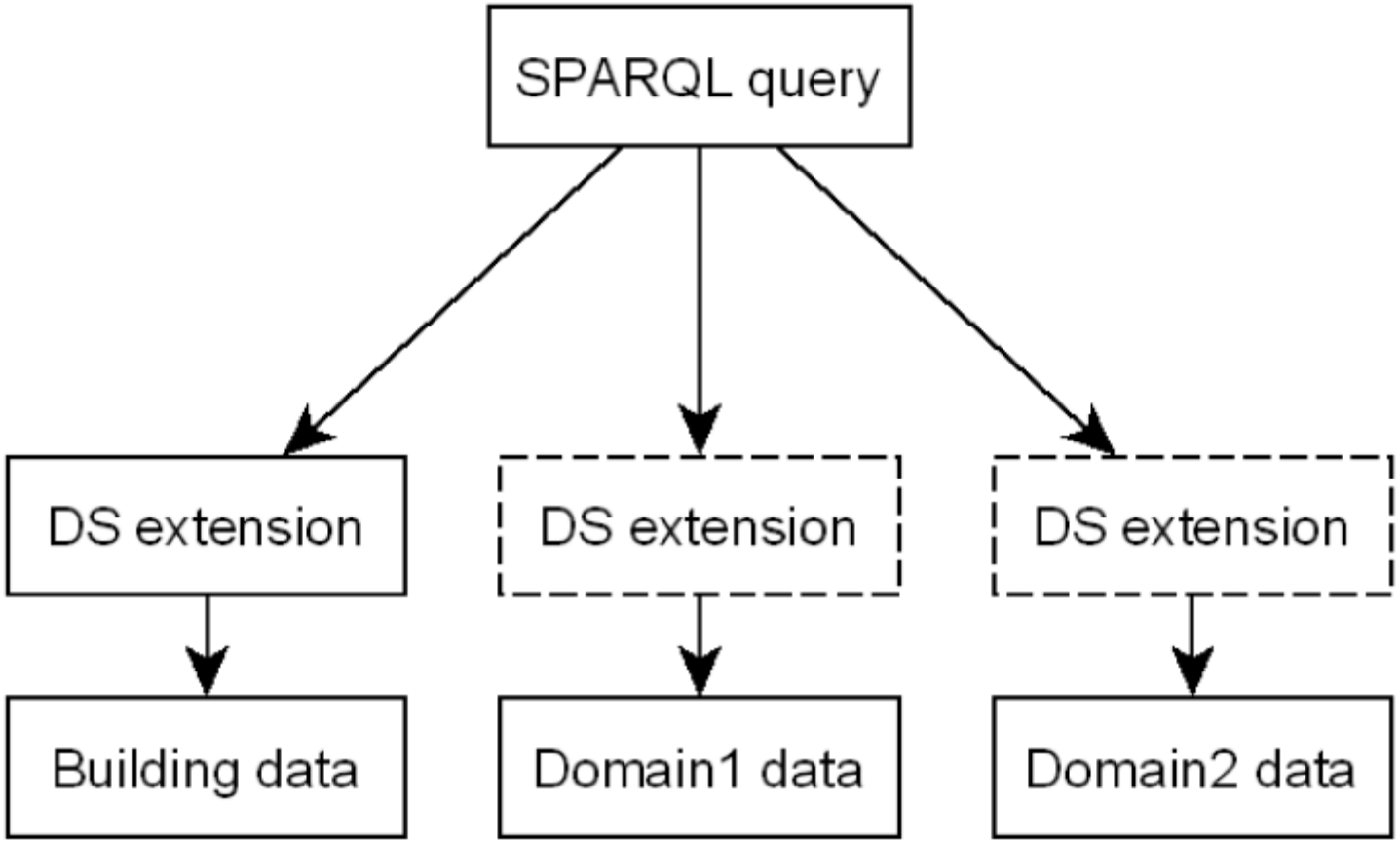
EXPRESS

MVDxml

SimpleBIM

Pieter Pauwels, Ana Roxin. SimpleBIM: from full ifcOWL graphs to simplified building graphs. Proceedings of the 11th European Conference on Product and Process Modelling. p.11-18.

BIMSPARQL



Chi Zhang, Jakob Beetz. Querying Linked Building Data Using SPARQL with Functional Extensions. Proceedings of the 11th European Conference on Product and Process Modelling.

GAP

SimpleBIM

BIMSPARQL

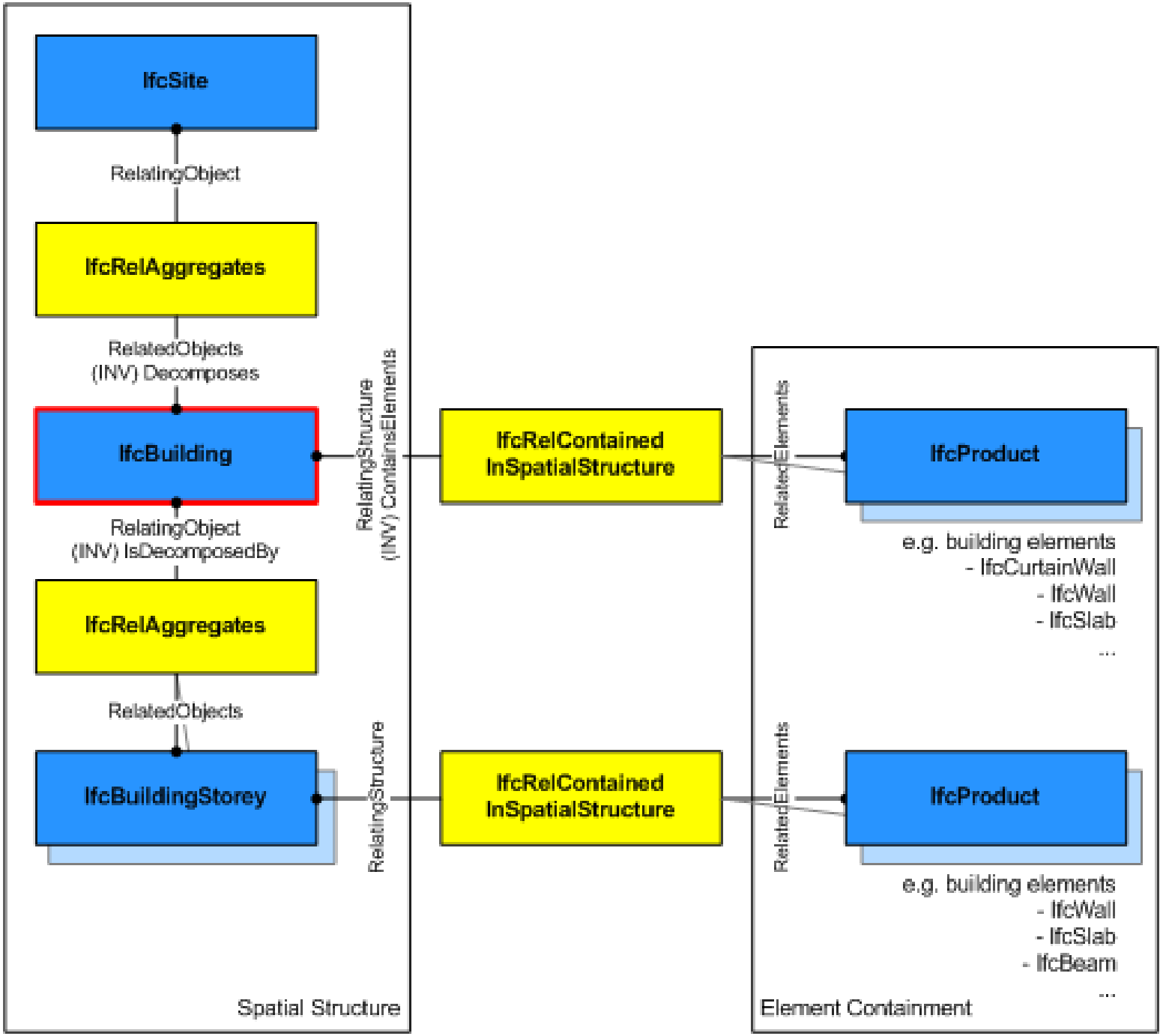
IFC-SPF

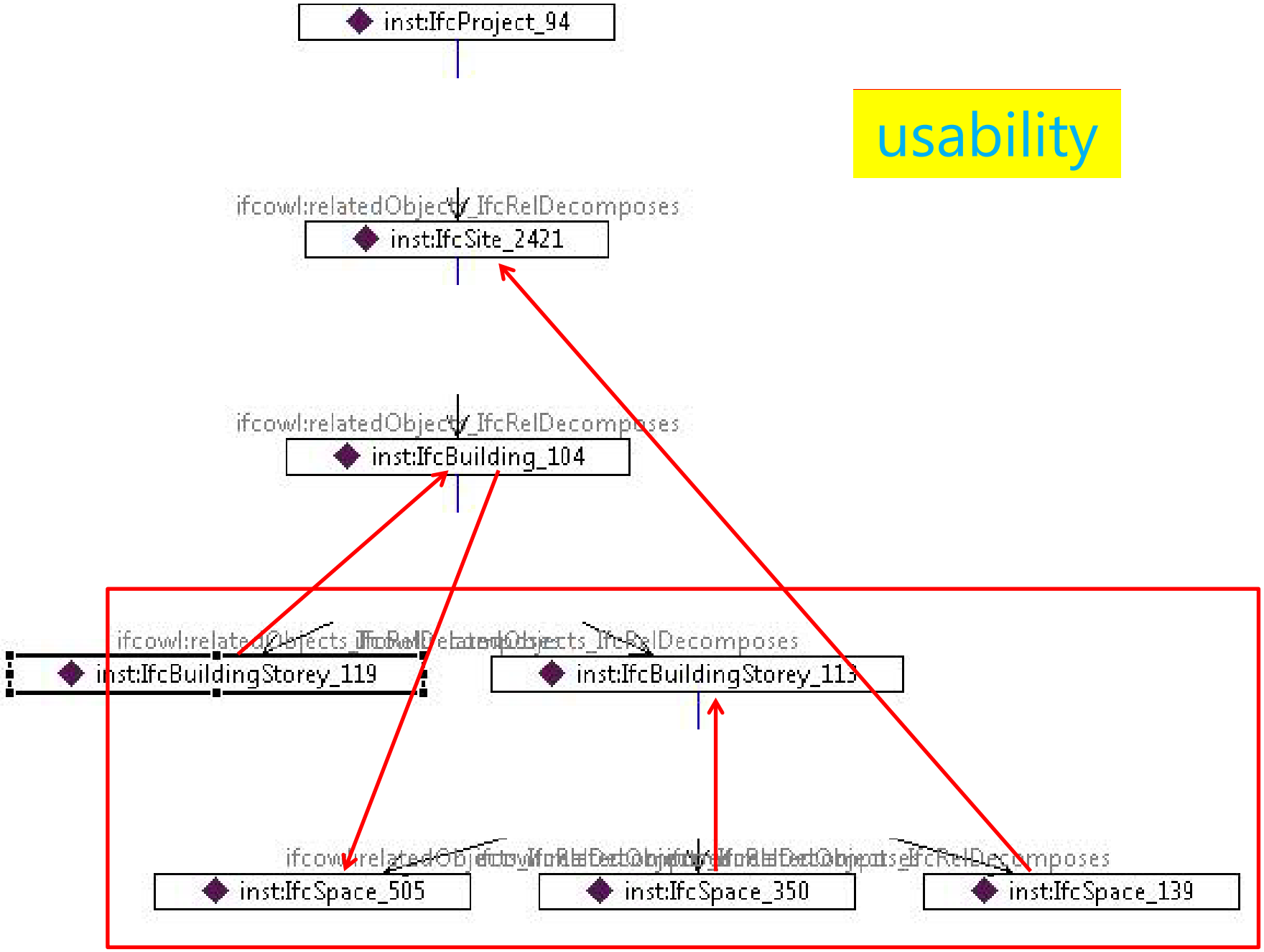
MVD subset

Simple Query Access

EXPRESS

MVDxml


IfcSite
RelatingObject
IfcRelAggregates
RelatedObjects
(INV) Decomposes
IfcBuilding
RelatingObject
(INV) IsDecomposedBy
IfcRelAggregates
RelatedObjects
IfcBuildingStorey
RelatingStructure
(INV) ContainsElements
RelatingStructure
IfcRelContained
InSpatialStructure
IfcRelContained
InSpatialStructure
Spatial Structure
RelatedElements
IfcProduct
e.g. building elements
- IfcCurtainWall
- IfcWall
- IfcSlab
...
RelatedElements
IfcProduct
e.g. building elements
- IfcWall
- IfcSlab
- IfcBeam
...
Element Containment


inst:IfcProject_94
usability
ifcowl:relatedObjects_IfcRelDecomposes
inst:IfcSite_2421
ifcowl:relatedObjects_IfcRelDecomposes
inst:IfcBuilding_104
inst:IfcBuildingStorey_119
inst:IfcBuildingStorey_113
inst:IfcSpace_505
inst:IfcSpace_350
inst:IfcSpace_139

1. MVD background
2. The motivation
3. Publishing (static) versus querying (dynamic)
4. White paper

Outline

GAP

SimpleBIM

BIMSPARQL

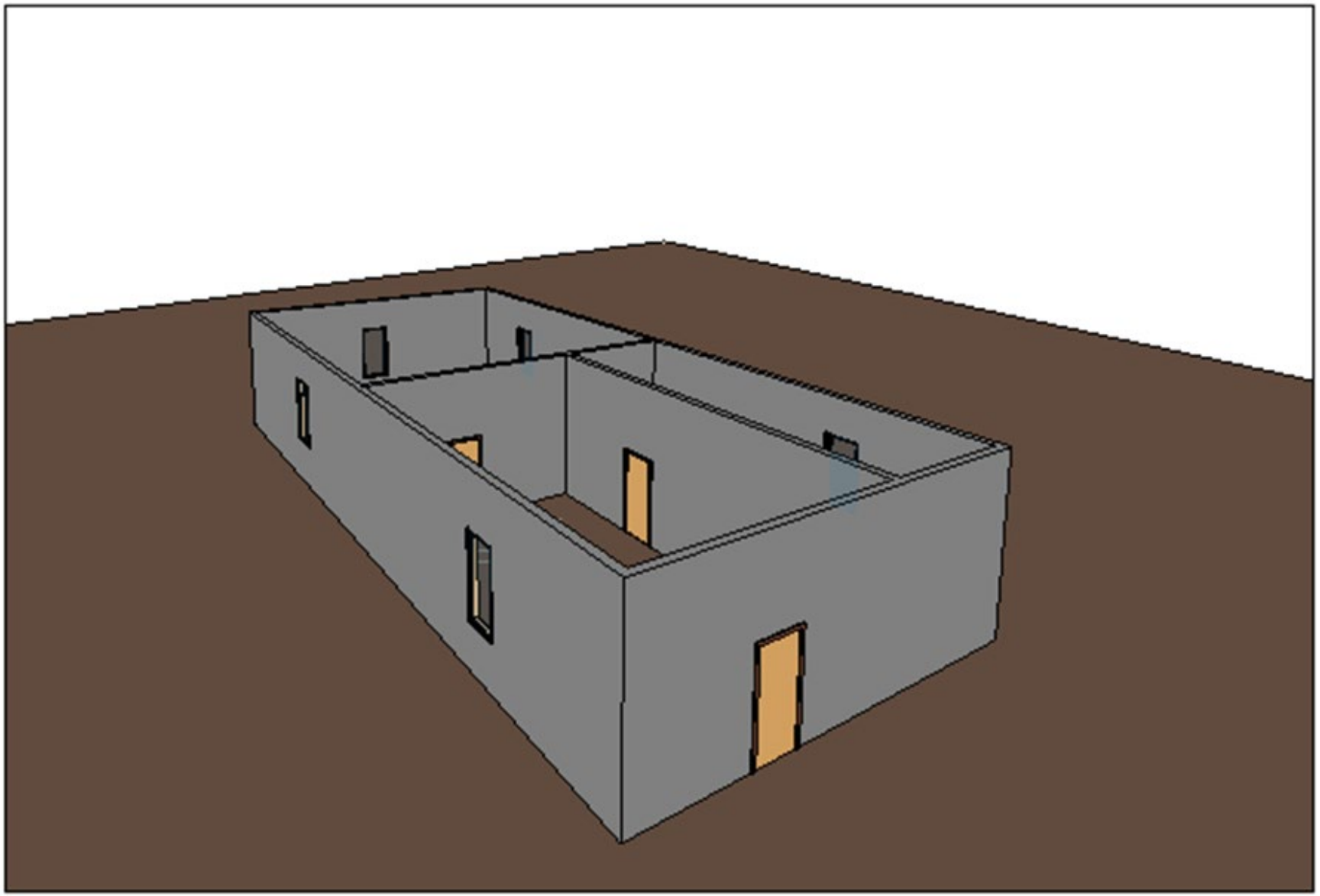
IFC-SPF

MVD subset

Simple Query Access

EXPRESS

MVDxml



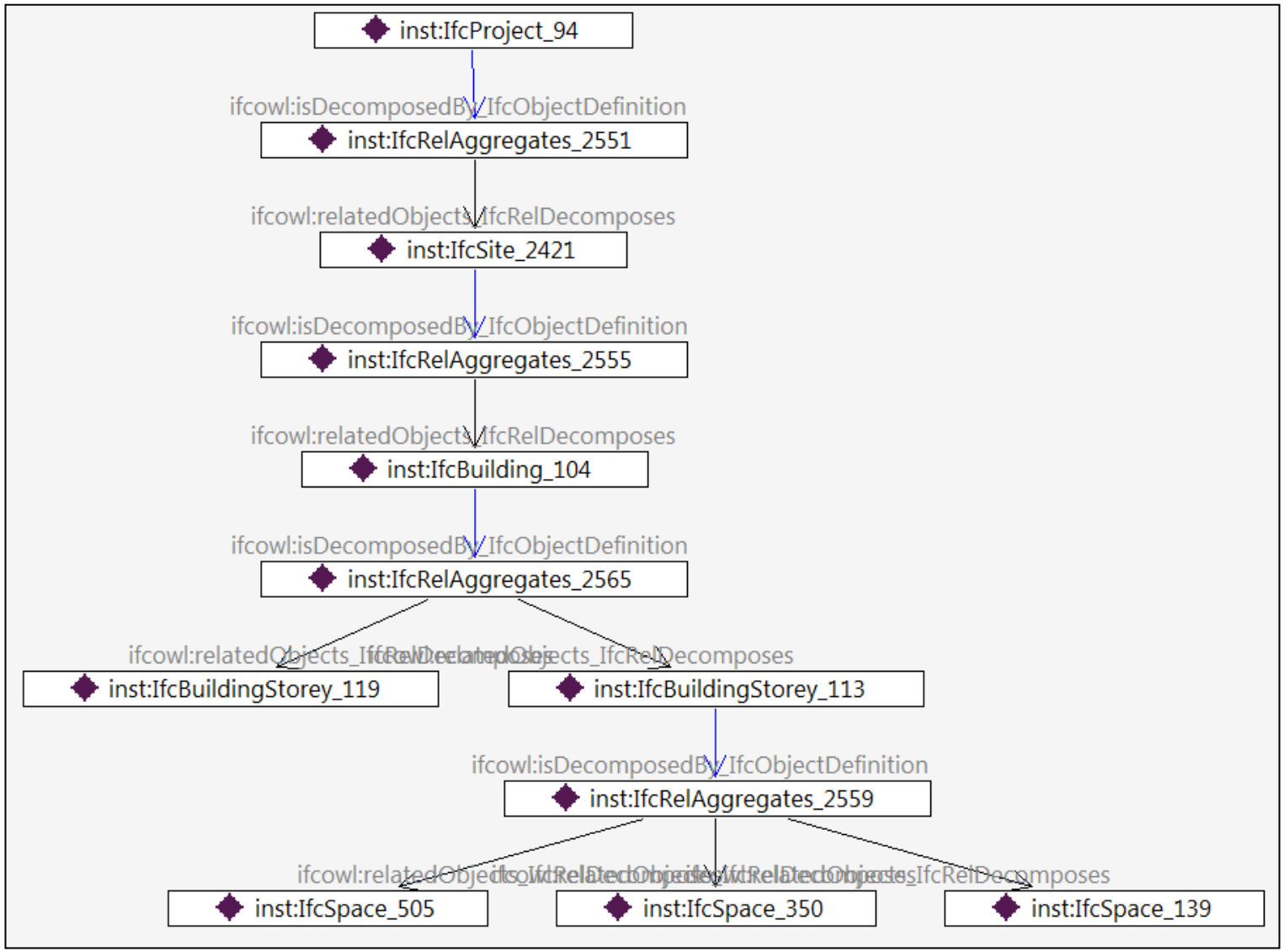
Statistics of the test file

- File size: 767kB
- Triple count: 10,173 distinct
- Class instances: 4222 (5535)
  - 233 / 4222 ifcowl:IfcRelationships
  - 686 / 4222 list:OWLList
  - 417 / 686 ifcowl:IfcLengthMeasure_List
  - 764 / 4222 expr:STRING

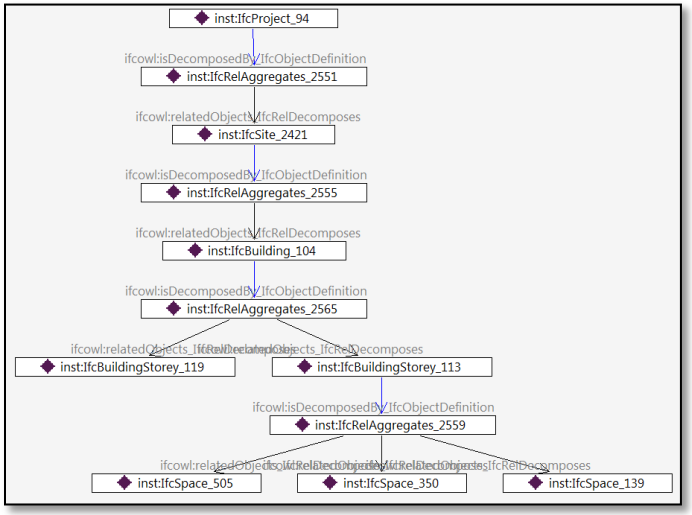
Simplification strategy

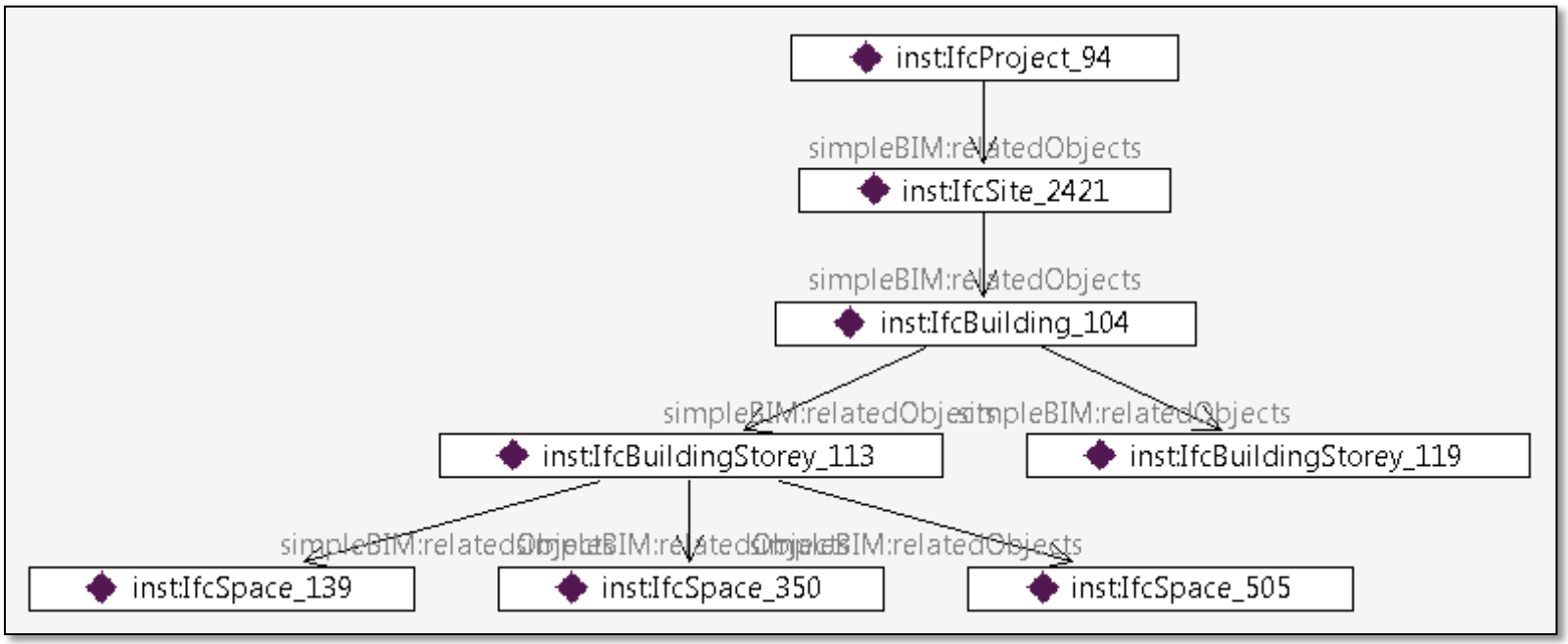
1. • Removing geometric information
2. • Unwrapping data types
3. • Rewriting properties
4. • IfcRelationship instances

Simplifying IfcRelationship instances

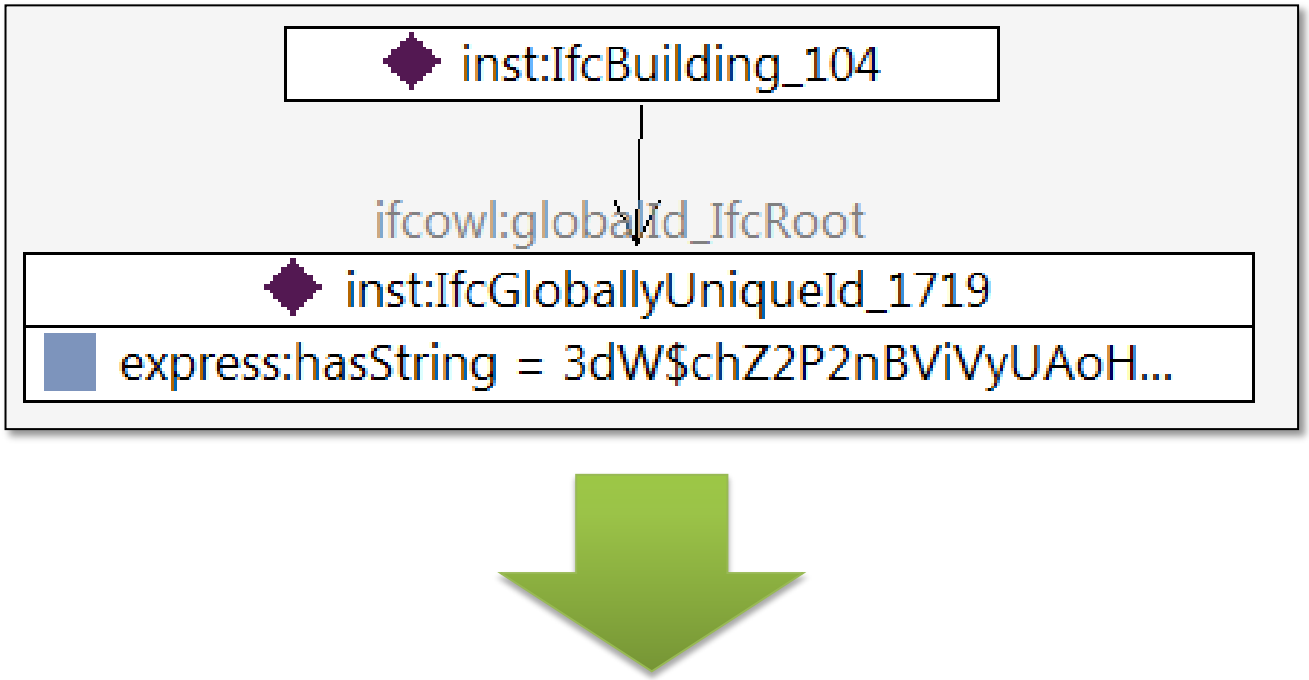


Simplifying IfcRelationship instances

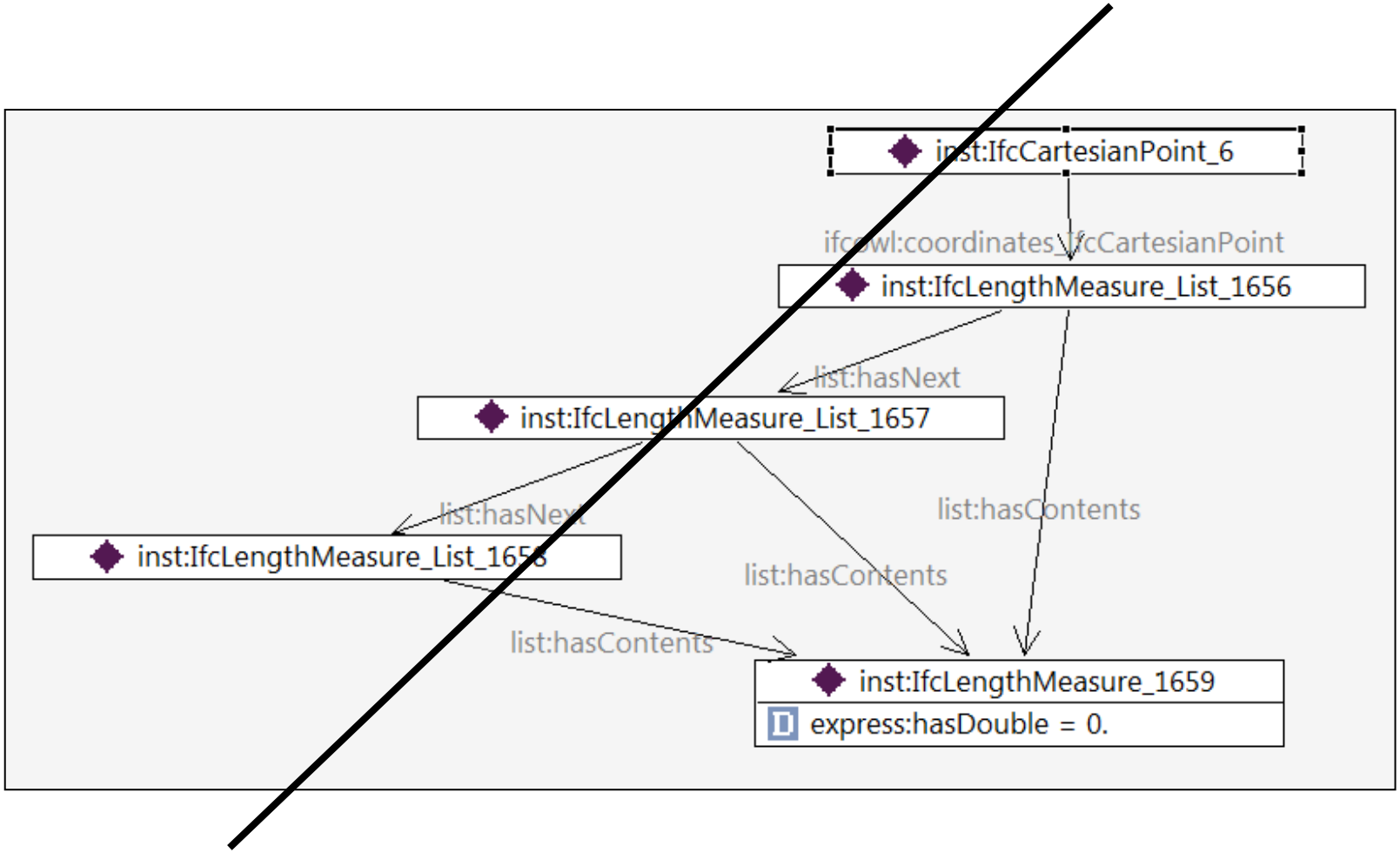




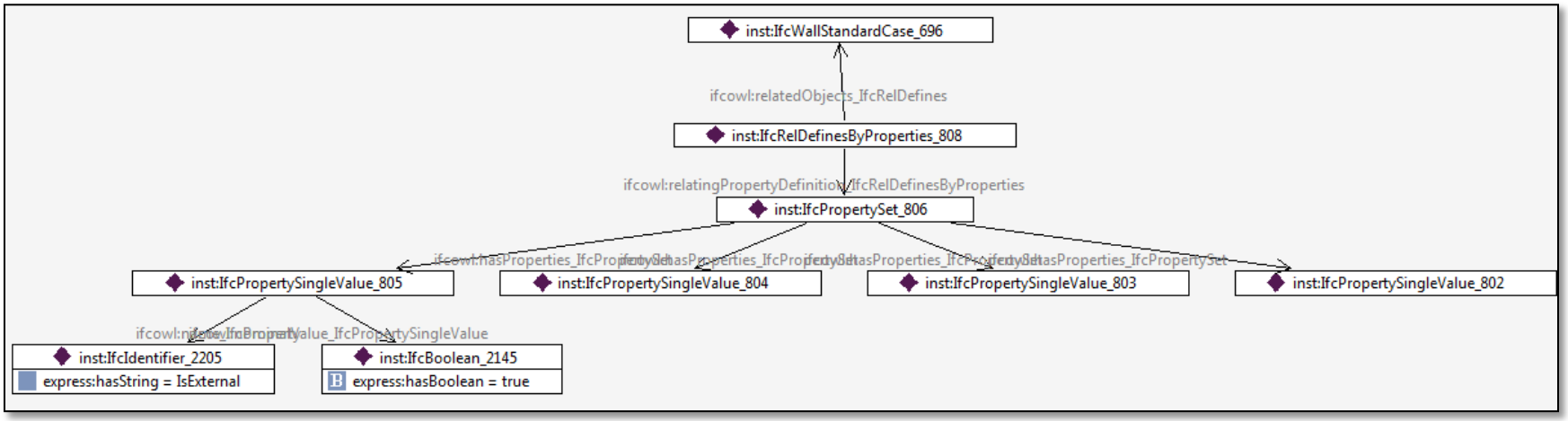
Unwrapping data types



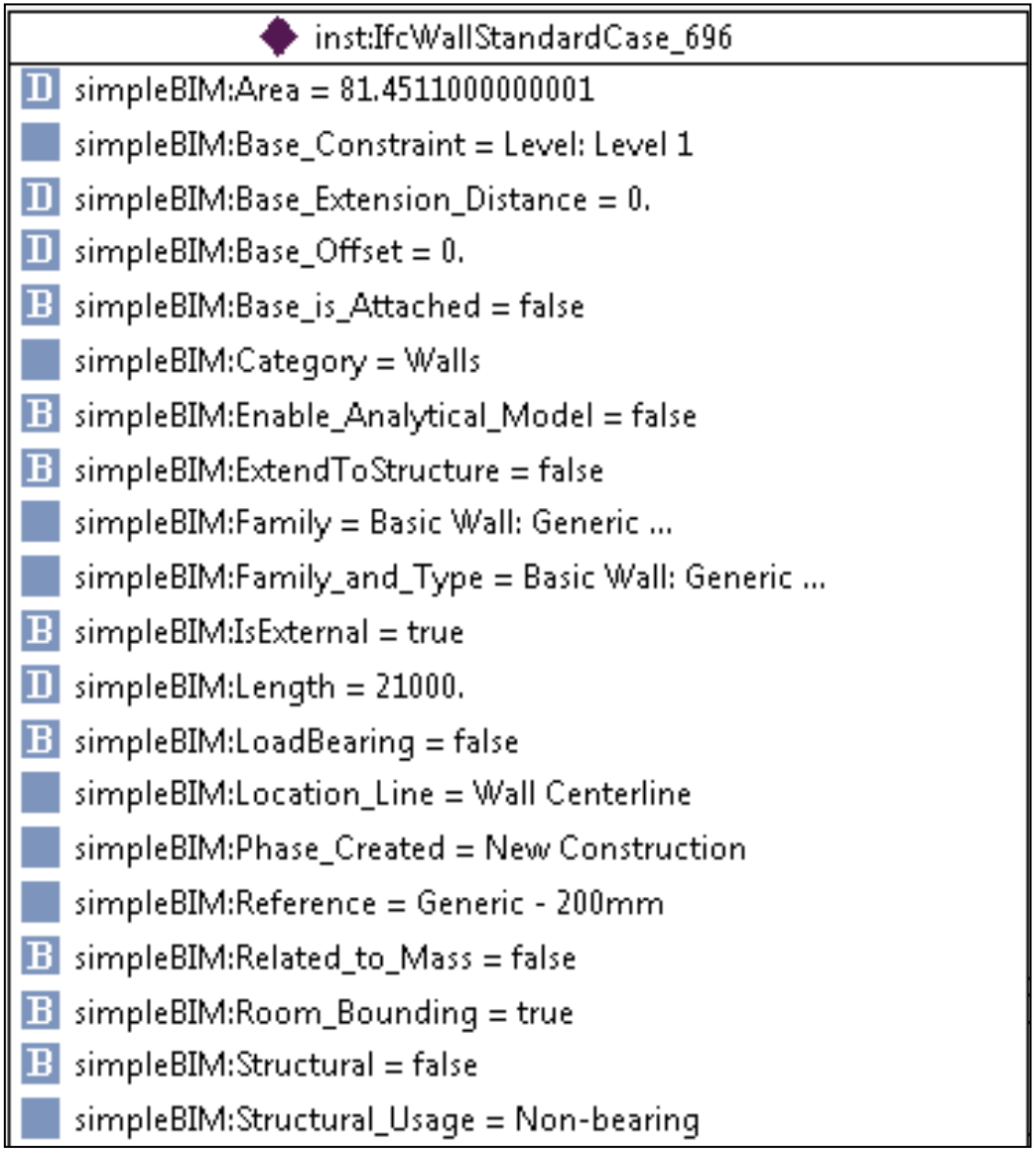
Removing geometric information



Rewriting PSETs and property values



Rewriting PSETs and property values



Results (1)

1. Removal of geometric information

- 10,173 triples to 6,927 triples
- 767kb to 476kb
- 31% (file size) – 38% (triple count)

2. Unwrapping data types

- 3,897 triples
- 279kb
- 41% (file size) – 44% (triple count)

Results (2)

3. Rewriting properties

- 1,630 triples
- 112kb
- 58% (file size) – 59% (triple count)

4. IfcRelationship instances

- 1,339 triples
- 83kb
- 18% (file size) – 26% (triple count)

Results (3)

Model	File size		Number of triples	
	ifcOWL	simpleBIM	ifcOWL	simpleBIM
1	767kb	83kb	10 173	1 339
2	16,7MB	1029kb	225 135	16 836

Average reduction of 91,58%

Average reduction of 89%

REDUCTION TO:
8,5% of file size
10,3% of triple count

GAP

SimpleBIM

BIMSPARQL

IFC-SPF

MVD subset

Simple Query Access

EXPRESS

MVDxml

1. MVD – linked data: motivation
2. From mvdXML to Selection Rules
3. Matchmaking with IfcDoc
4. White paper

Outline

buildingSMART
International home of openBIM

Document id	Title	Organisation /Author	Date	Status
WP_LDWG1	LDWG MVD White Paper	LDWG – Pieter Pauwels LDWG – Peter Muigg LDWG – Seppo Törmä LDWG – Kris McGlinn LDWG – Matthias Weise LDWG – Yong-Cheol Lee LDWG – Ana Roxin LDWG – Goncal Costa LDWG – Nam Vu Hoang	29 Mar. 2016	DRAFT

White Paper:
On the relation between Model View Definitions (MVDs) and Linked Building Data

Executive summary

This white paper outlines the proposals from the LDWG on how technologies and approaches that are common to the domains of Semantic Web, Linked Data, and the Web of Data (hereafter jointly called 'Linked Data') are related to Model View Definitions (MVDs)

S. Törmä, Nam Vu Hoang
A. Roxin, T. Mendes de Farias

Executive summary

1. Model View Definitions
2. Linked Data
3. Technical: handling MVDs with Linked Data technologies
4. Industrial use cases
5. Conclusions

intro

outro

K. McGlinn, M. Weise
(W3C LBD, SWIMing)

3.1. Publishing strategy: translate the outcome of a regular MVD subset generation process into an RDF graph

In this approach, the regular MVD technology stack is used, allowing to generate an mvdXML file that specifies which kind of data should be included in an IFC file in order to allow the Exchange Requirements as specified in an IDM. Resulting IFC files can then also be validated against this mvdXML using the IfcDoc tool.

In addition to the above, the validated MVDs (subset IFCs) can also be published as subset RDF graphs. As a result, there are as many RDF graphs published as there are MVDs defined in the IDM. Eventually, there are thus a number of RDF graphs that represent subset IFC models. As they are readily available, they can easily be reused for various purposes by directly loading or querying them.

The advantages of this approach are:

- High reusability, as data is readily available in RDF
- Speed, as data is readily available in RDF

The disadvantages of this approach are:

- Cumbersome process (export to IFC, generate MVD subset in IfcDoc, validate, convert to RDF, publish as subset) to eventually generate a subset
- Hard maintenance, as multiple models need to be managed which are actually all related to a master model, but are not managed as such.

3.2. Filtering strategy: represent mvdXML files in semantic rules or queries and do on-demand MVD subset generation.

In this approach, the full IFC model is directly published as an RDF graph and the mvdXML files, which specify particular MVDs, are expressed in semantic rules (SWRL) or queries (SPARQL). By doing so, the entire subset generation process that was proposed in 3.1 (IFC -> IfcDoc MVD -> IFC -> RDF) becomes available in a complete semantic web form. Namely, the semantic rules or queries can be combined on demand with the full IFC model in RDF, so that an MVD-compliant subset becomes available on demand as well. Besides the availability of generic semantic web tools (query and reasoning engines mainly), this approach could also be motivated by the ability to take into account external linked data (e.g. material data or geographical data).

GAP

SimpleBIM

BIMSPARQL

IFC-SPF

MVD subset

Simple Query Access

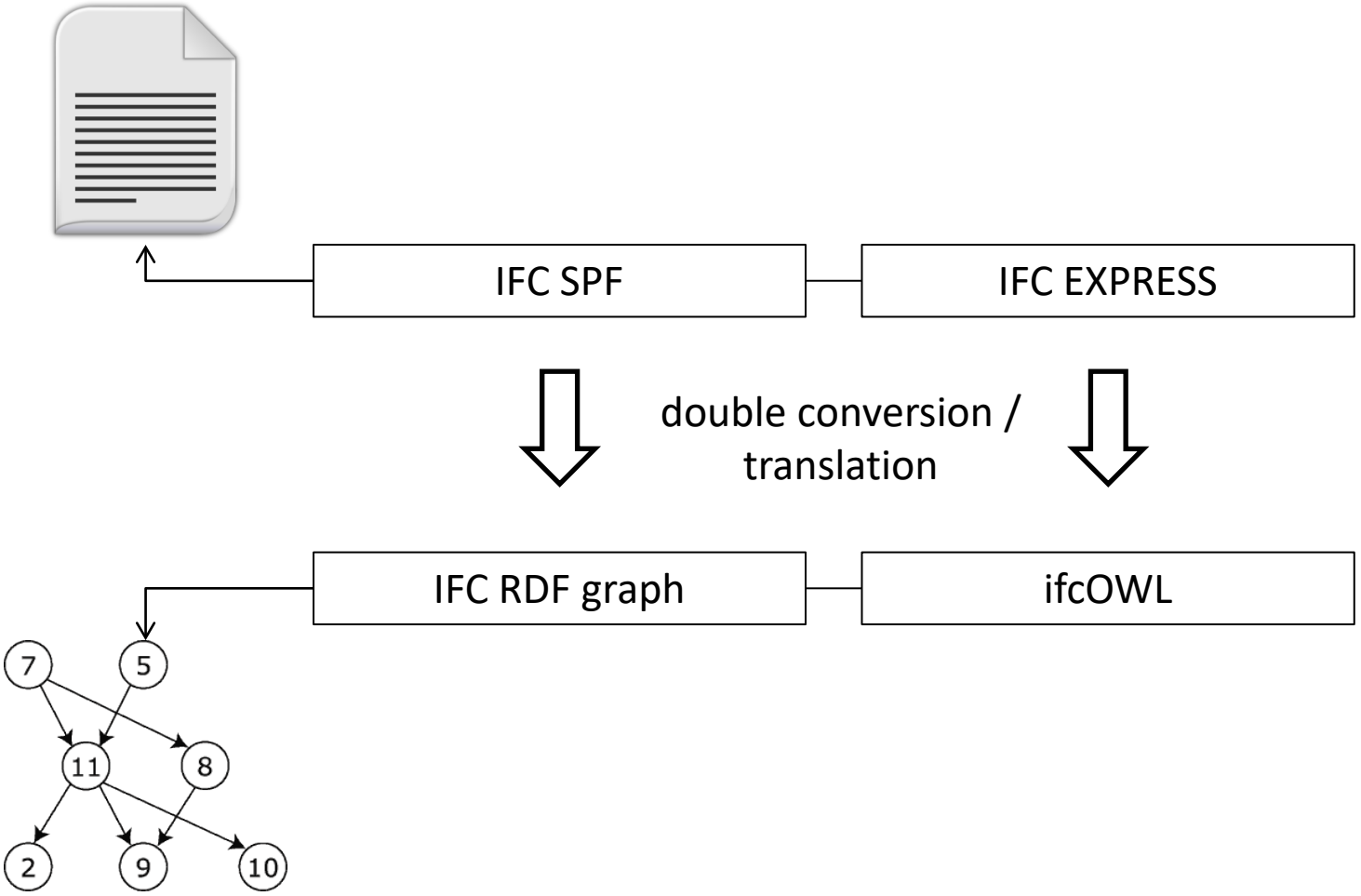
EXPRESS

MVDxml

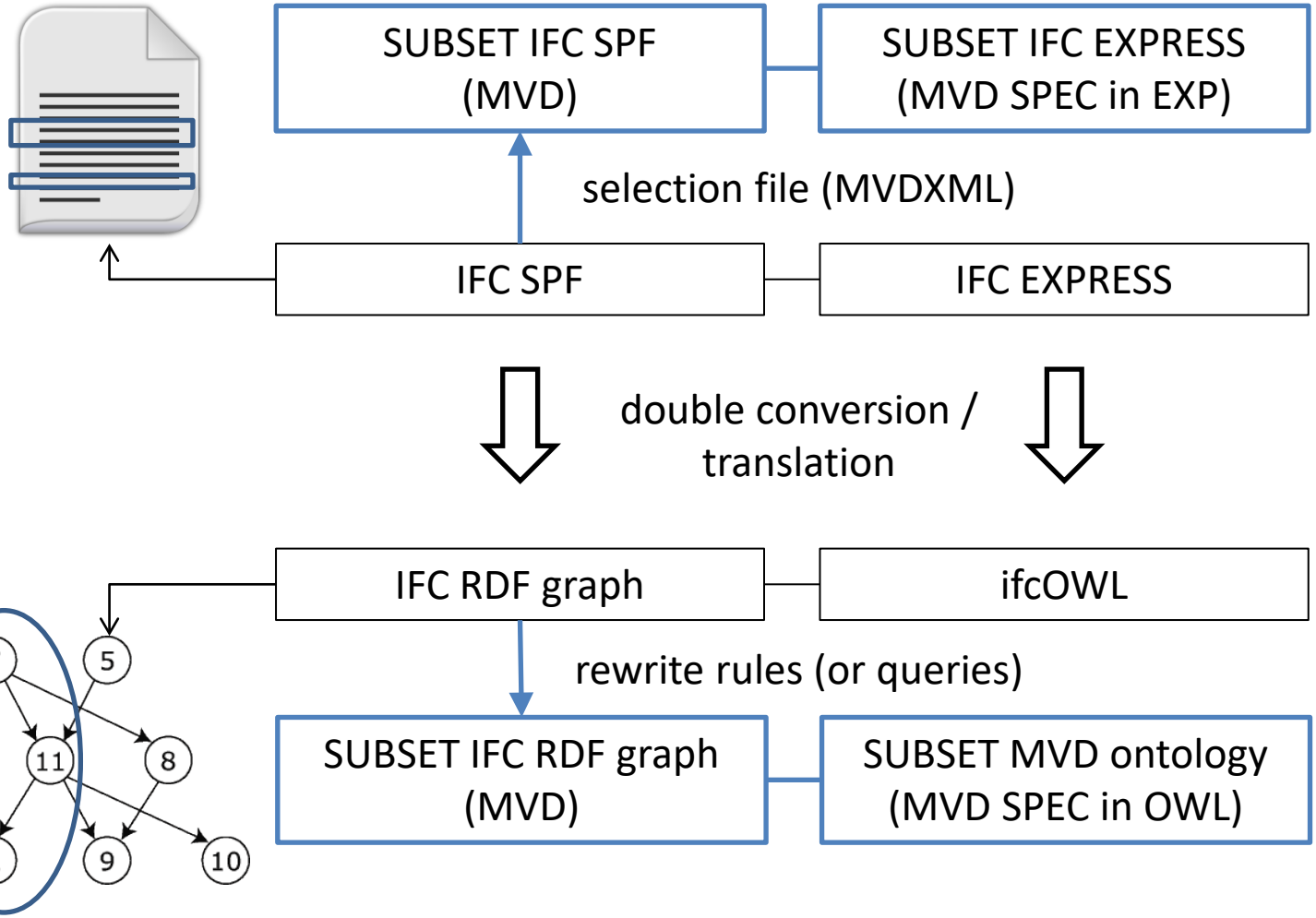
Thank you

Pieter Pauwels – pipauwel.pauwels@ugent.be

What is available



What do we want to do with this



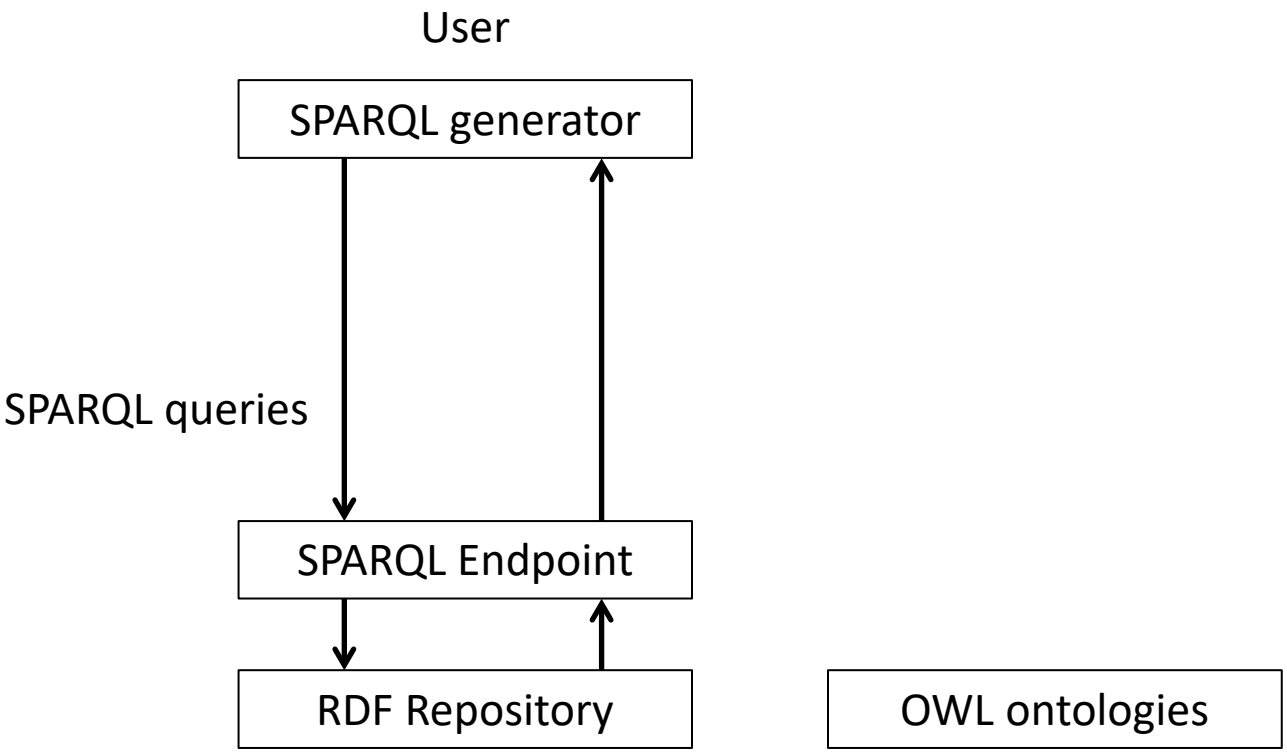
ROUTE 2 - APPLYING THE MVD GRAPHS TO AN IFC MODEL

Option 1 – SPARQL queries

```
[prefixes]

CONSTRUCT
{
        [OUTPUT GRAPH AS DESIRED]
}
WHERE
{
        ?a rdf:type ifcowl:IfcSpace .
        ?a rdf:type ifcowl:IfcRoot .
        ?a ifcowl:GlobalId ?b .
        ?b rdf:type ifcowl:IfcGloballyUniqueId .
        ?b ifcowl:has_string ?c .
        ?a ifcowl:Name_of_IfcRoot ?d .
        ?d rdf:type ifcowl:IfcLabel .
        ?d ifcowl:has_string ?e .
        ?a ifcowl:Description_of_IfcRoot ?f .
        ?f rdf:type ifcowl:IfcText .
        ?f ifcowl:has_string ?g .
}
```

SPARQL Schematic overview



Remarks

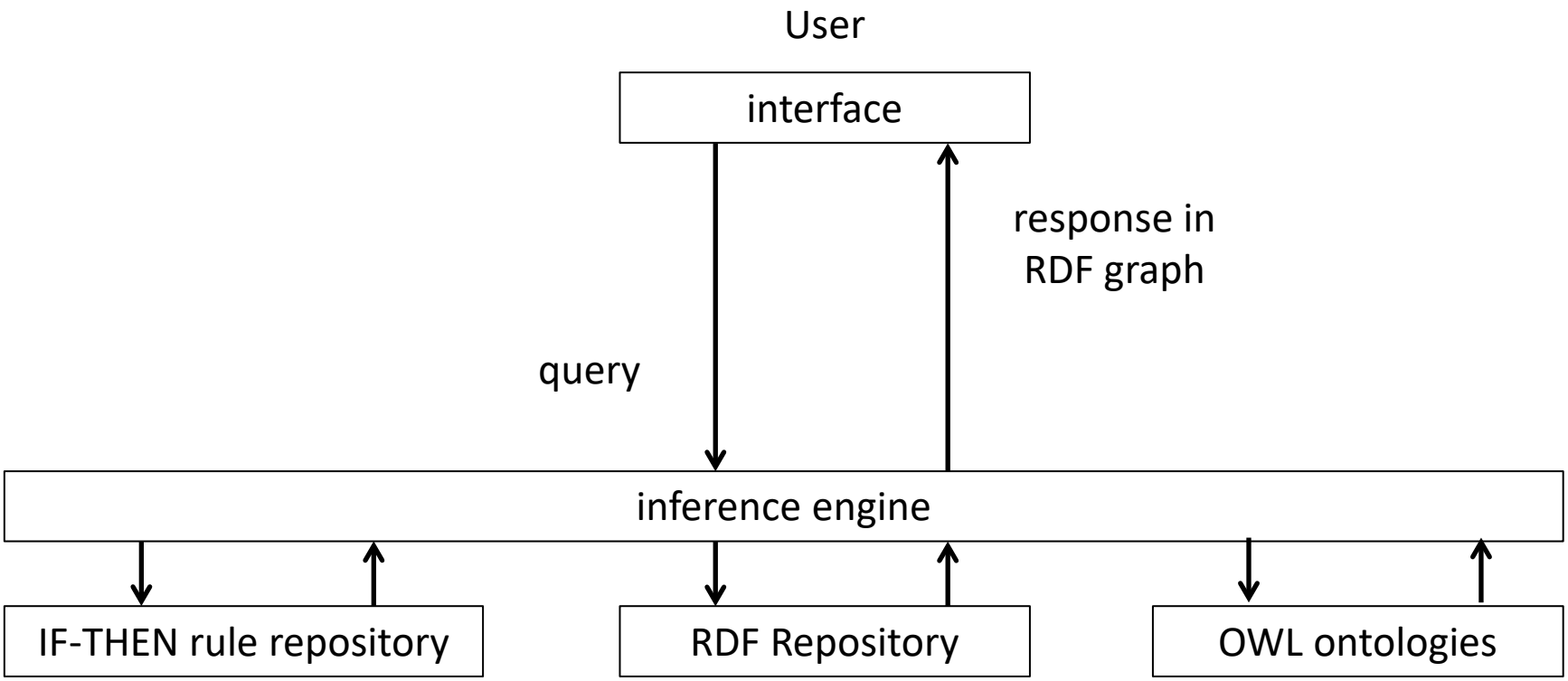
- SPARQL queries are fired **one by one**
- They need to be processed by the receiver (application or end user), so there is **no real ‘single output subset graph**’ available
- **Not easy** to write SPARQL queries
- Need to maintain **SPARQL endpoint**
- Compliance with a **stable ontology** needed

Option 2 – IF-THEN graph rules

```
[prefixes]

{
        ?a rdf:type ifcowl:IfcSpace .
        ?a rdf:type ifcowl:IfcRoot .
        ?a ifcowl:GlobalId ?b .
        ?b rdf:type ifcowl:IfcGloballyUniqueId .
        ?b ifcowl:has_string ?c .
        ?a ifcowl:Name_of_IfcRoot ?d .
        ?d rdf:type ifcowl:IfcLabel .
        ?d ifcowl:has_string ?e .
        ?a ifcowl:Description_of_IfcRoot ?f .
        ?f rdf:type ifcowl:IfcText .
        ?f ifcowl:has_string ?g .
}
=>
{
        [OUTPUT GRAPH AS DESIRED]
}
```

Rules schematic overview



Remarks

- One can **combine as many rules** as one wishes in one ‘inference run’
- The receiver (application or end user) receives the result as a ‘**single output subset graph**’
- The individuals are still in the original namespace (so they have the same URIs / are identical). The relations, class attributions and potential additional statements can be placed in a separate namespace. But the **link to the original** is always maintained through the URIs of the individuals.
- **Performance** can change a lot, depending on the amount of resources and rules that is loaded by the reasoning engine.
- **Not easy** to write rules
- Compliance with a **stable ontology** needed

ROUTE 2 - THE RESULT IN IFCRDF (DIRECTLY)

```
@prefix :      <http://www.buildingsmart-tech.org/ifcOWL#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix mvd:   <http://linkedbuildingdata.net/ifc/rules/20150630_144046/> .
@prefix dce:   <http://purl.org/dc/elements/1.1/> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix inst:  <http://linkedbuildingdata.net/ifc/instances20150609_015154#> .
@prefix log:   <http://www.w3.org/2000/10/swap/log#> .

inst:IfcLabel_273230
                a  mvd:IfcLabel ;
                mvd:has_string  "Ghent" .

inst:IfcLabel_273233
                a  mvd:IfcLabel ;
                mvd:has_string  "Belgium" .

inst:IfcBuilding_36
                a       mvd:IfcBuilding ;
                mvd:BuildingAddress  inst:IfcPostalAddress_35 .

inst:IfcPostalAddress_35
                a           mvd:IfcPostalAddress ;
                mvd:Country  inst:IfcLabel_273233 ;
                mvd:Region   inst:IfcLabel_273231 ;
                mvd:Town     inst:IfcLabel_273230 .

inst:IfcLabel_273231
                a  mvd:IfcLabel ;
                mvd:has_string  "Flanders" .
```

MVD building location - IfcBuilding

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix mvd:   <http://linkedbuildingdata.net/ifc/rules/20150630_144046/> .
@prefix ifcowl: <http://www.buildingsmart-tech.org/ifcOWL#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix inst:  <http://linkedbuildingdata.net/ifc/instances20150703_192529#> .

inst:IfcBuilding_36
        a     mvd:IfcObjectDefinition , mvd:IfcBuilding ;
        mvd:IsDecomposedBy  inst:IfcRelAggregates_266702 .

inst:IfcRelAggregates_266702
        a      mvd:IfcRelAggregates ;
        mvd:RelatedObjects_of_IfcRelAggregates          inst:IfcBuildingStorey_40 ,
                inst:IfcBuildingStorey_43 , inst:IfcBuildingStorey_47 , inst:IfcBuildingStorey_51 .

inst:IfcBuildingStorey_43      a      mvd:IfcSpatialElement .

inst:IfcBuildingStorey_47      a      mvd:IfcSpatialElement .

inst:IfcBuildingStorey_51      a      mvd:IfcSpatialElement .

inst:IfcBuildingStorey_40      a      mvd:IfcSpatialElement .
```

Other subsets tested

- IfcBuilding location
- IfcElement type selection + Identity
- Spatial composition / decomposition
- SI units used for IfcProject
- Selection of specific property values
- IfcBuildingStorey elevation
- IfcSite location
- IfcProject context
- Object typing
- Spatial containment
- IfcSpace classification
- Placement
- Mapped geometry / Body geometry / Body BREP geometry / Body SweptSolid geometry
- Space boundaries and connection geometry

The CONCEPT in an MVDXML

```
<ConceptRoot uuid="83c2915f-08f1-4027-a32c-23fd3ee588b6" name="" applicableRootEntity="IfcBuildingStorey">
    <Concepts>
      <Concept uuid="64802040-8b87-4ed2-87bc-b9cc16a587df" name="Identity" override="false">
          <Template ref="f7cb0a08-fee7-4cc0-85fe-43f6a6a0bce1" />
          <Requirements />
          <Rules />
      </Concept>
      <Concept uuid="9e903641-5cb8-4e64-a479-4d4619c2ddc1" name="with Property names" override="false">
          <Template ref="d5449a96-e8a9-48f0-bc9a-c3829f5fc668" />
          <Requirements />
          <Rules>
            <TemplateRule Parameters="Name=Pset_BuildingStoreyCommon;PropertyName=Reference;" />
            <TemplateRule Parameters="Name=Pset_BuildingStoreyCommon;PropertyName=EntranceLevel;" />
            <TemplateRule Parameters="Name=Pset_BuildingStoreyCommon;PropertyName=AboveGround;" />
          </Rules>
      </Concept>
    </Concepts>
</ConceptRoot>
```