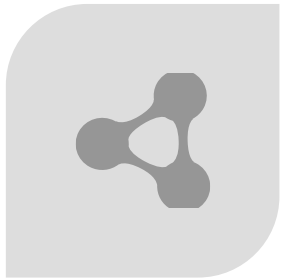
SHACL is for LBD what mvdXML is for IFC

M.Sc. Jyrki Oraskari, M.Sc. Madhumitha Senthilvel, and Dr. Jakob Beetz

Introduction



HOW TO **MINIMISE DATA LOSS AND ERRORS** WHEN DATA IS CONVERTED INTO LINKED BUILDING DATA

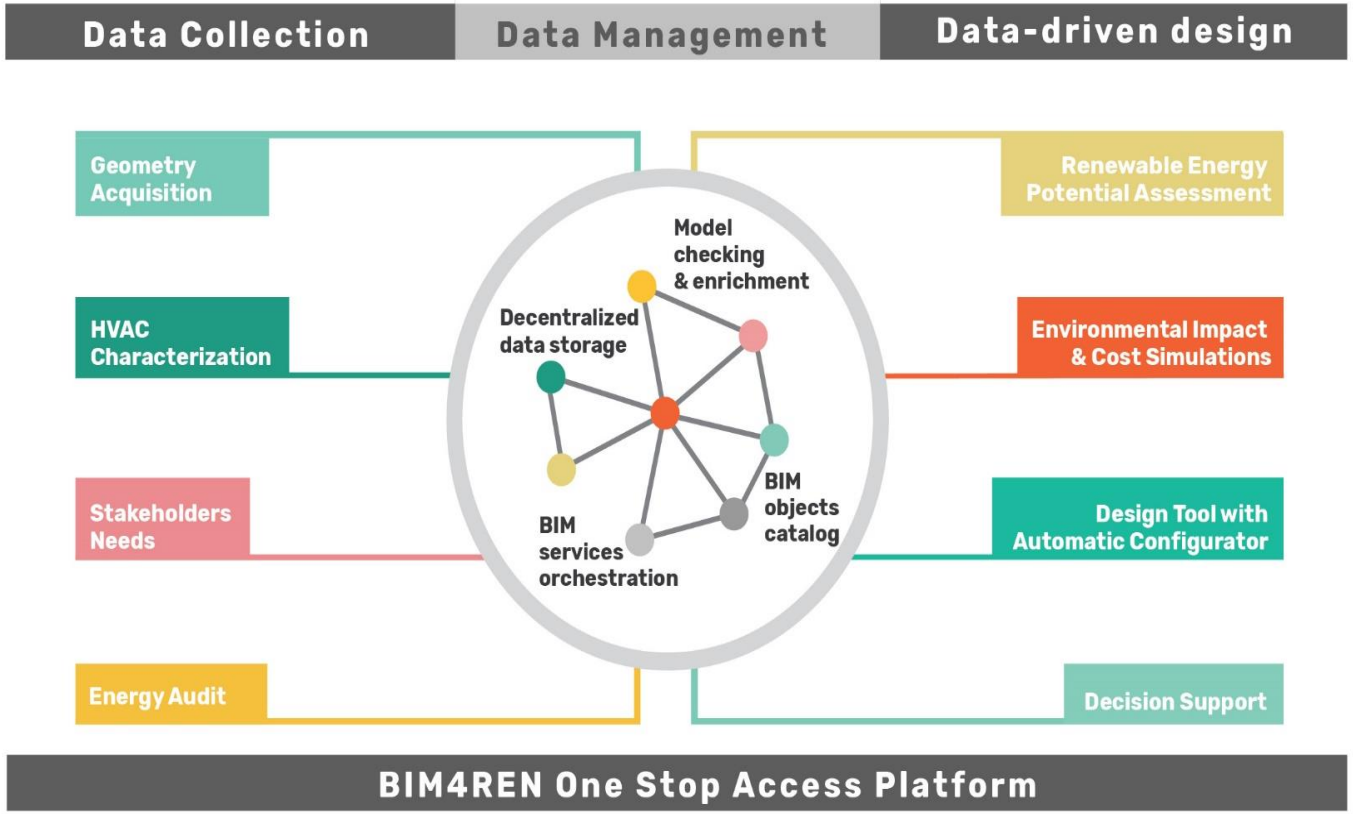
HOW UNIT TESTS CAN IMPROVE THE **LINKED DATA QUALITY** WHEN IT IS BOUGHT INTO THE SYSTEM

HOW **MODEL VIEW DEFINITION** CAN BE FORMED

Image: http://www.w3.org/RDF/icons/rdf_flyer.64,

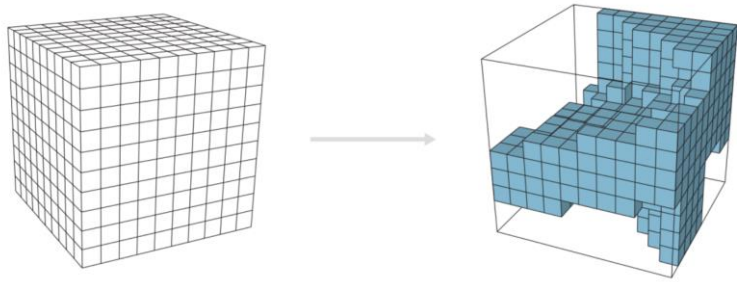
BIM for Renovation: BIM4Ren

- Online framework for renovation-oriented BIM (H2020)
- Interconnecting heterogeneous information sources and tools represented as services
- Toolchains for real-world renovation scenarios
- 23 partners across Europe
- Practitioners, Developers, Researchers, SMEs
- One Stop Access Platform (Common Data Environment)



Model View Definition (MVD)

"Currently, **MVD**s are encoded in a format called **mvdXML**, and define allowable values at particular attributes of particular data types. For example, an MVD may require that a wall provide a fire rating" buildingSMART International



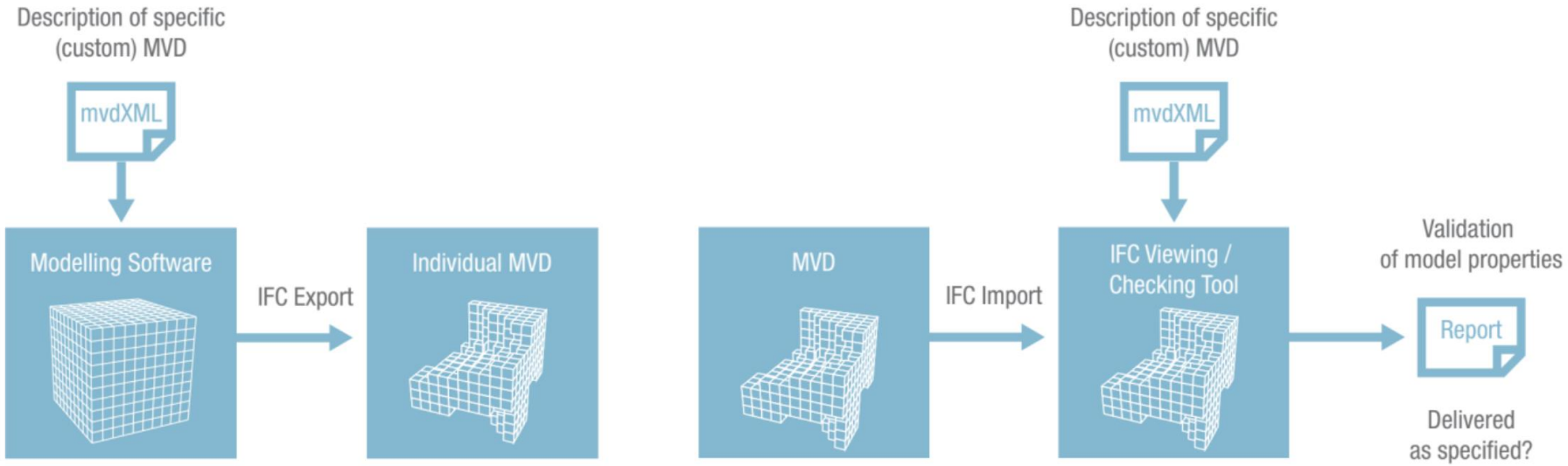


Image source: Baldwin 2017, The BIM-Manager: A Practical Guide for BIM Project Management

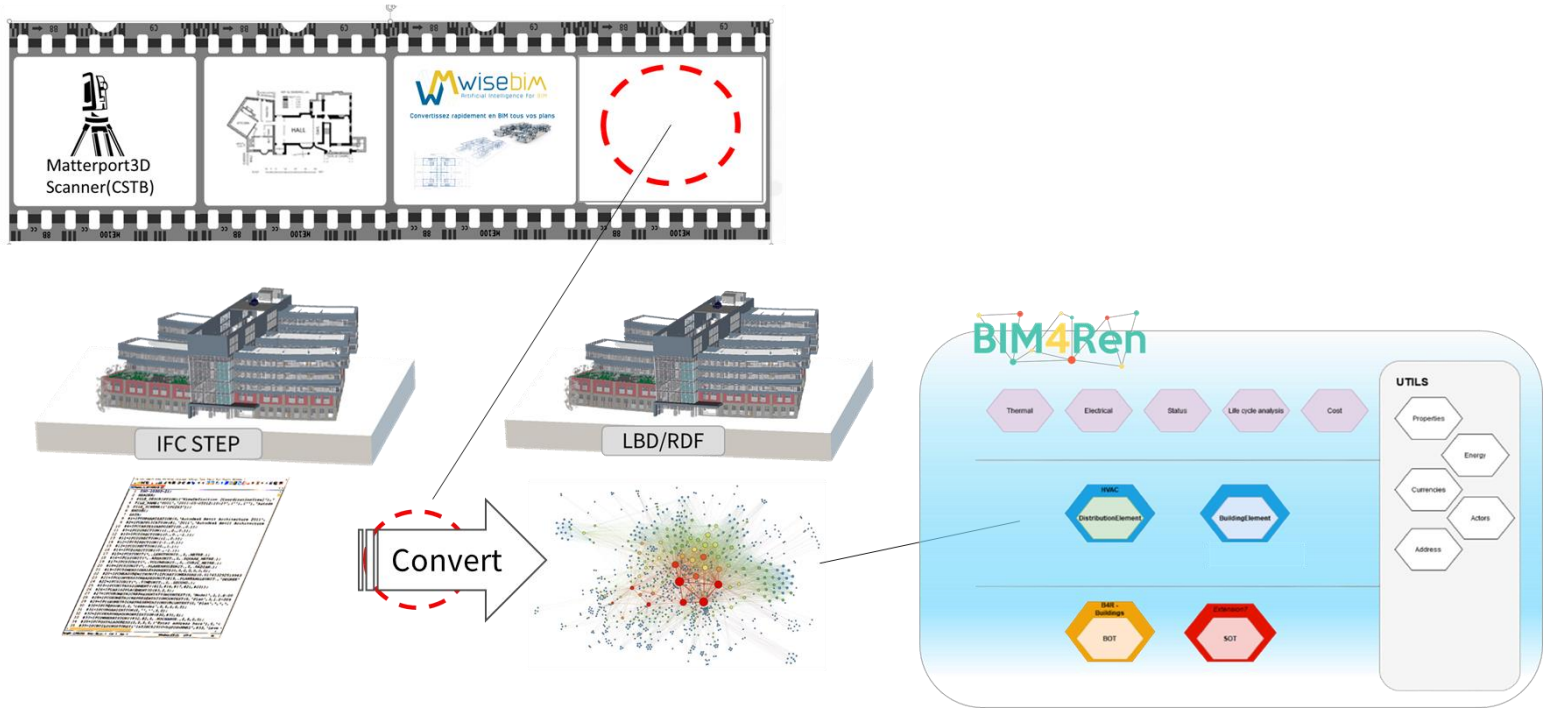
Shapes Constraint Language (SHACL)

SHACL is a general-purpose data validation language and specification from the World Wide Web Consortium (W3C)

Shapes Constraint Language (SHACL)

SHACL is a general-purpose data validation language and specification from the World Wide Web Consortium (W3C)

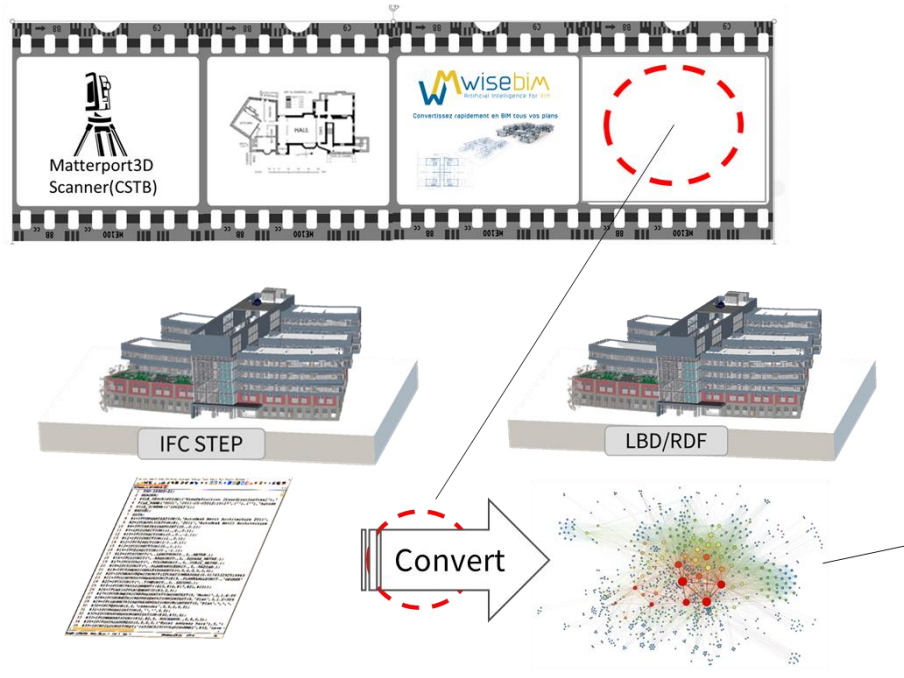
IFC to Linked Building Data (LBD)

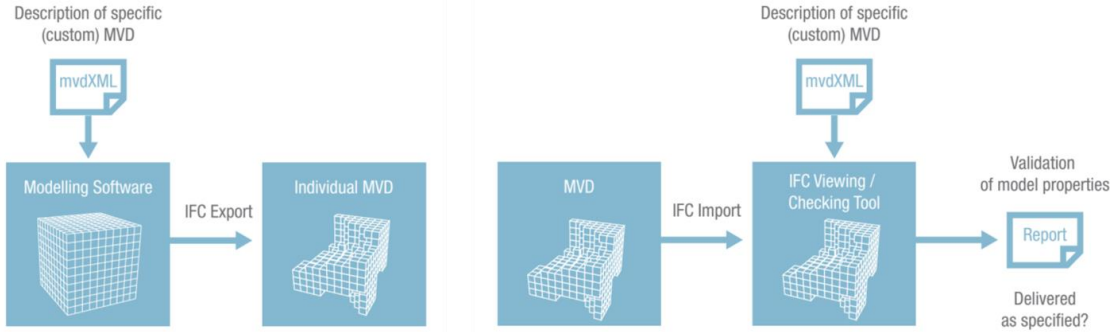


Shapes Constraint Language (SHACL)

SHACL is a general-purpose data validation language and specification from the World Wide Web Consortium (W3C)

IFC to Linked Building Data (LBD)





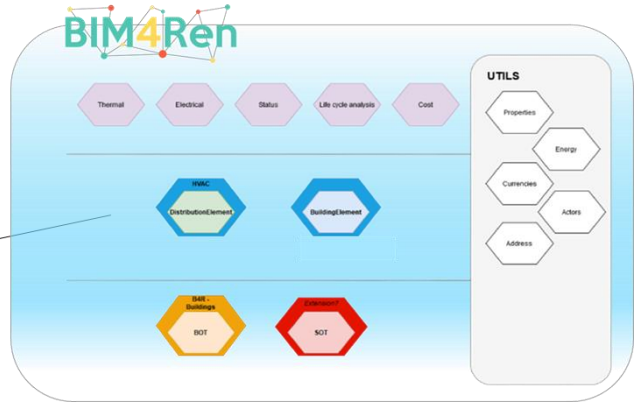
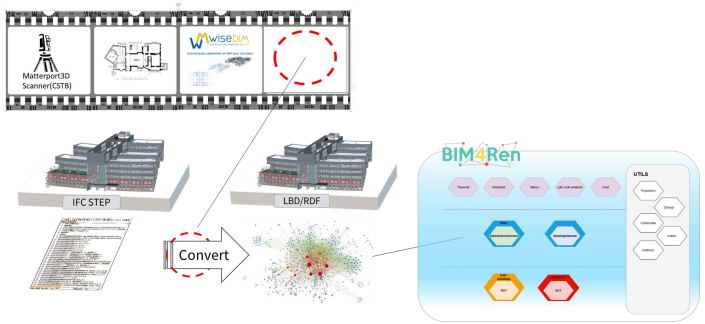
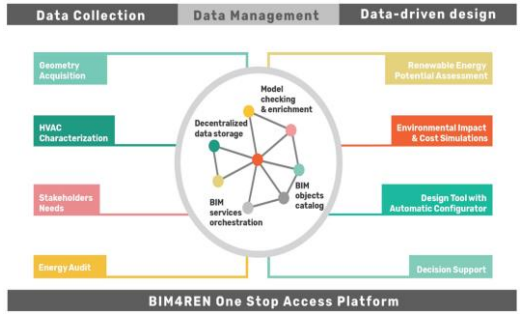
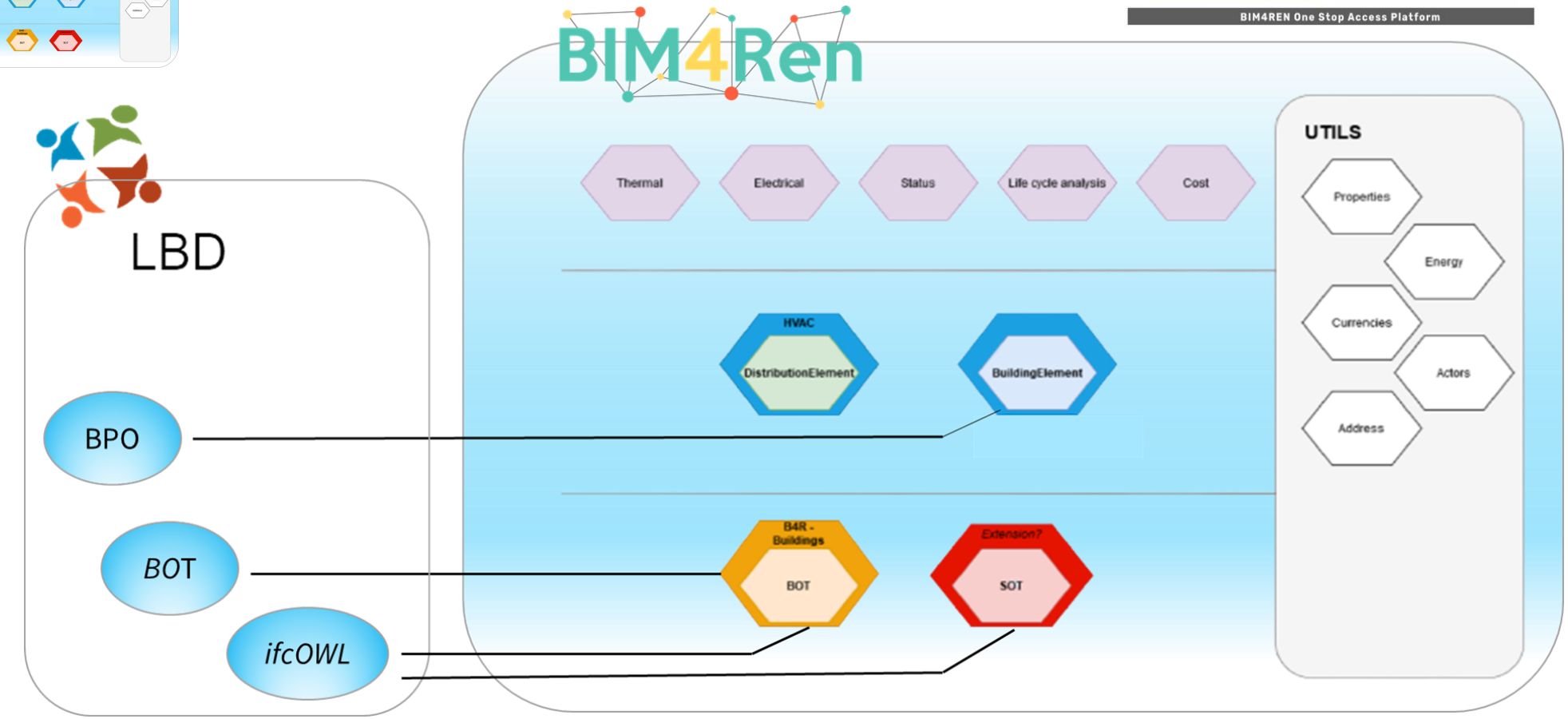


Image source: Baldwin 2017, The BIM-Manager: A Practical Guide for BIM Project Management

IFC to Linked Building Data (LBD)

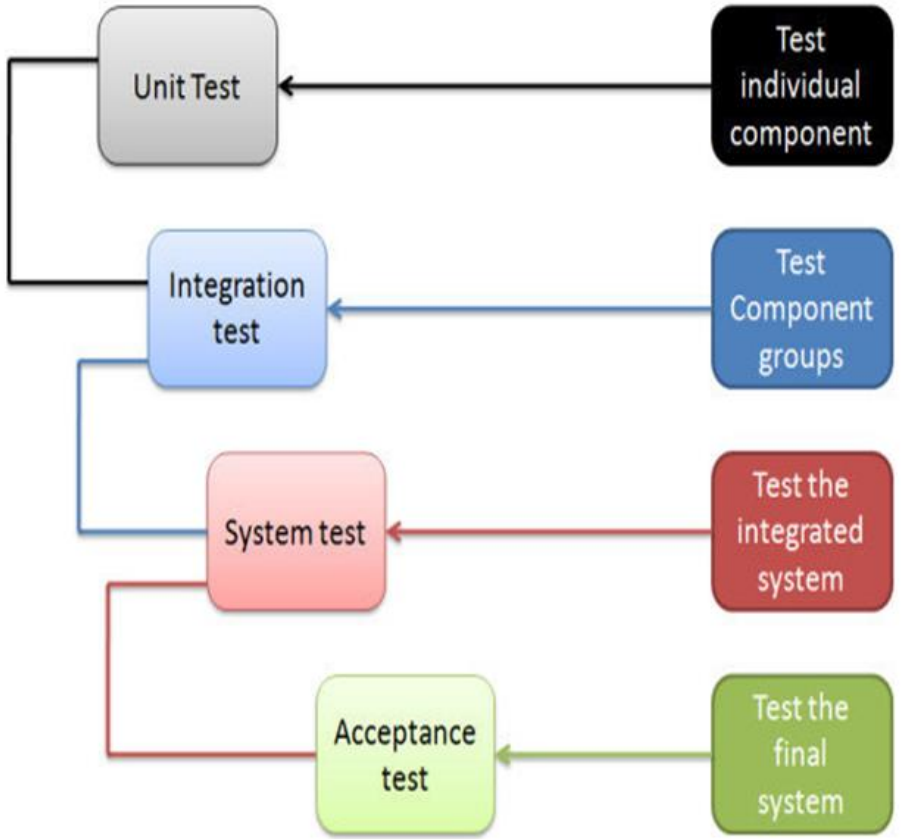






Unit testing in a software development

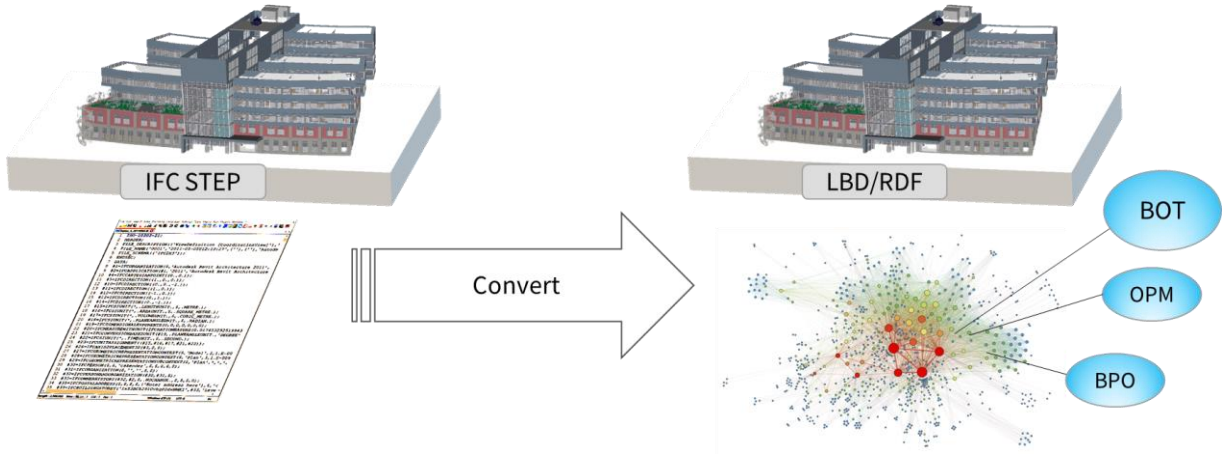
- to avoid unwanted errors in code **when introducing new features**.
- sample guided software design

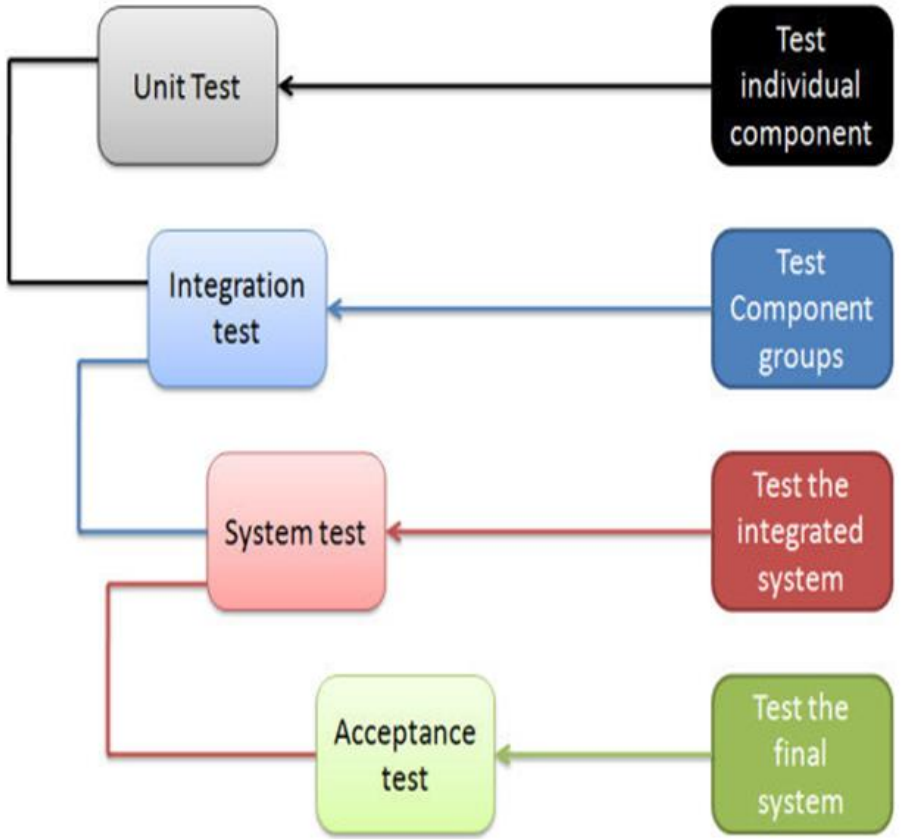


https://www.test-institute.org/Software_Testing_Levels.php

Unit testing in a software development

- to avoid unwanted errors in code **when introducing new features**.
- sample guided software design
- **the output-based unit tests**

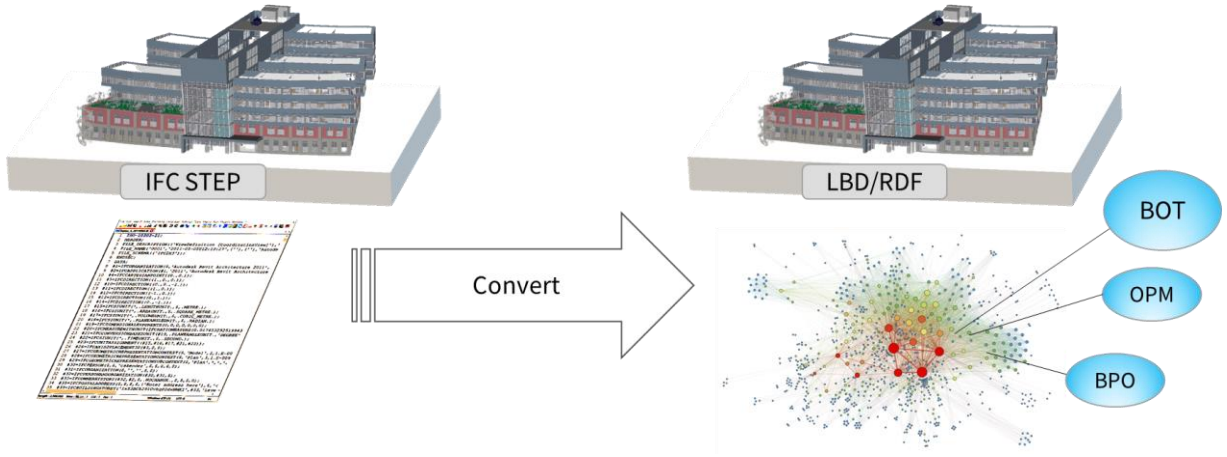




https://www.test-institute.org/Software_Testing_Levels.php

Unit testing in a software development

- to avoid unwanted errors in code **when introducing new features**.
- sample guided software design
- **the output-based unit tests**

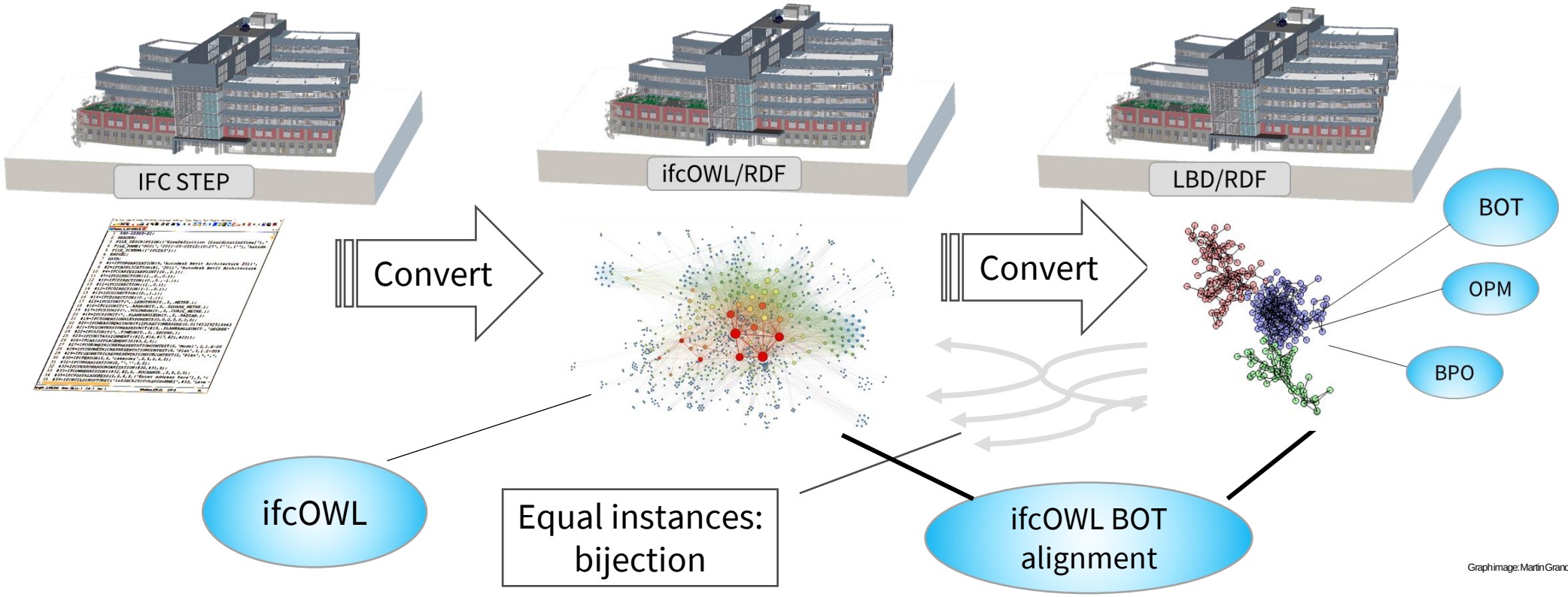


Requirements

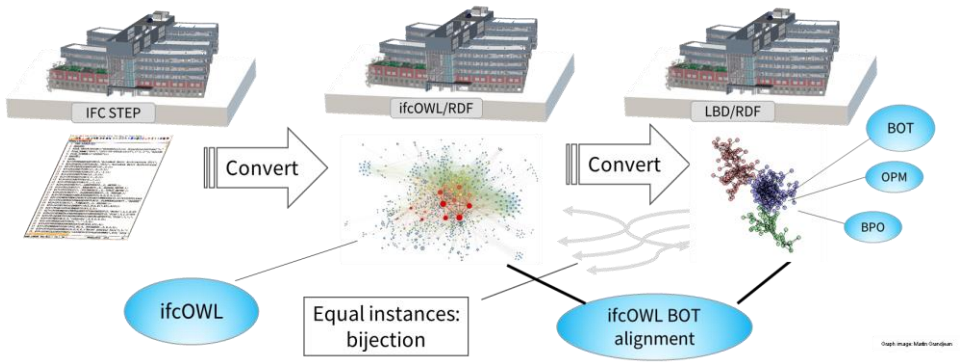
the unit tests validate the **ontology mapping** realization for individuals defined in the alignments

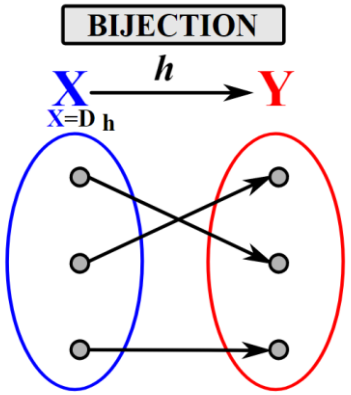
the unit test is to check that the translator keep **the relations and the attributes**

Unit tests for aligment



Unit tests for aligment





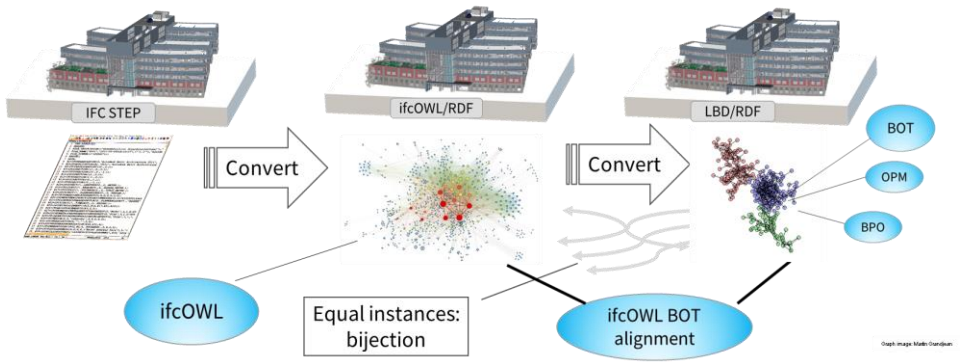
Wikimedia Commons: Lilyu

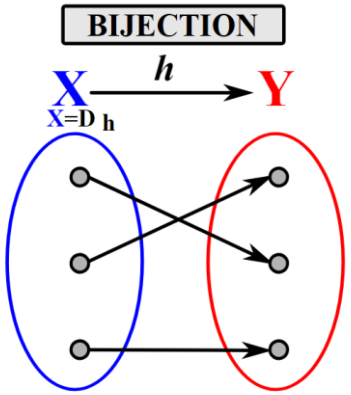
For selected subset of alignment:

- one-to-one correspondence (bijection):
  $x \mapsto f(y), y \mapsto f(x)$
- based on GUID

Yes/NO Image by Oberholster Venita from Pixabay
Split arrow: frlaticon

Unit tests for aligment





Wikimedia Commons: Lilyu

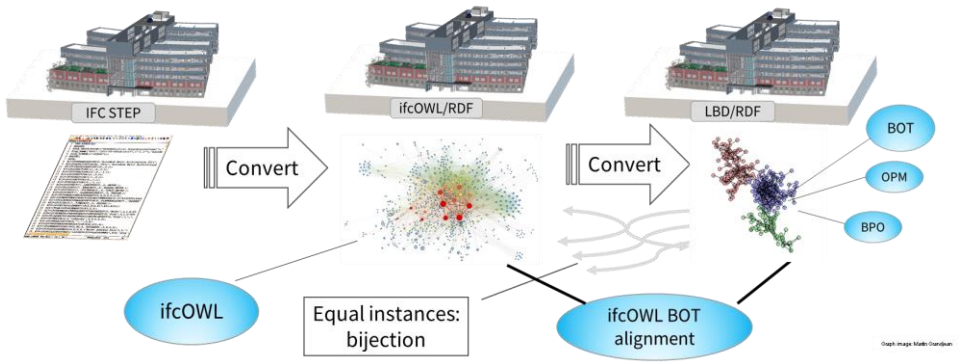
For selected subset of alignment:

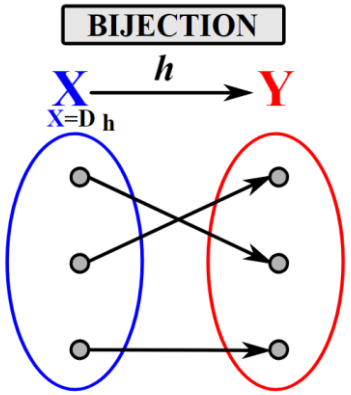
- one-to-one correspondence (bijection):
  $x \mapsto f(y), y \mapsto f(x)$
- based on GUID

SHACL is a general-purpose data validation language and specification from the World Wide Web Consortium (W3C)

Yes/NO Image by Oberholster Venita from Pixabay
Split arrow: frlaticon

Unit tests for aligment





Wikimedia Commons: Lilyu

For selected subset of alignment:

- one-to-one correspondence (bijection):
  $x \mapsto f(y), y \mapsto f(x)$
- based on GUID

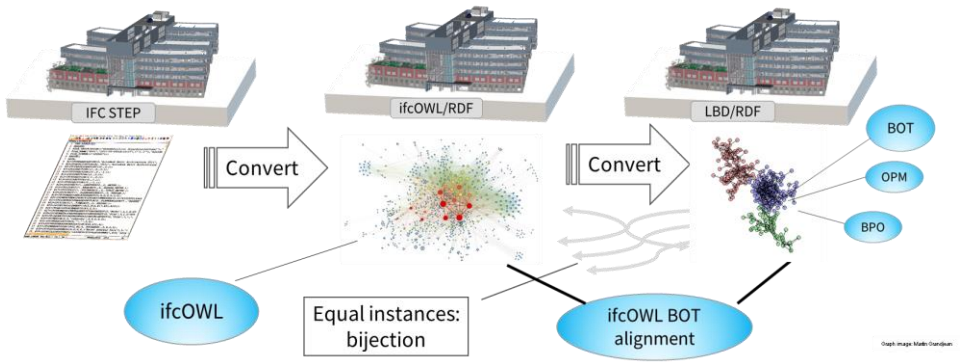
Listing 1. SHACL rule for checking BOT class alignment

```
1   inst:shapeBOTClasses a sh:NodeShape ;
2    sh:targetClass ifc:IfcSite, ifc:IfcBuilding,  ifc:Store;
3    sh:property [
4     sh:sparql [
5      a sh:SPARQLConstraint ;
6      sh:message "GUIDs of the aligment instances should match." ;
7      sh:prefixes inst:prefixes;
8      sh:select """
9       SELECT $this (?guid as ?value){
10       $this   ifc:globalId_IfcRoot/express:hasString ?guid .
11       FILTER NOT EXISTS { ?b props:globalIdIfcRoot_attribute_simple ?guid }
12       }
13     """
14     ]
```

SHACL is a general-purpose data validation language and specification from the World Wide Web Consortium (W3C)

Yes/NO Image by Oberholster Venita from Pixabay
Split arrow: frlaticon

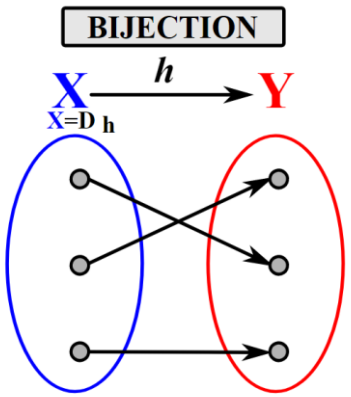
Unit tests for aligment



Listing 1. SHACL rule for checking BOT class alignment

```
1   inst:shapeBOTClasses a sh:NodeShape ;
2    sh:targetClass ifc:IfcSite, ifc:IfcBuilding,  ifc:Store;
3    sh:property [
4     sh:sparql [
5      a sh:SPARQLConstraint ;
6      sh:message "GUIDs of the aligment instances should match." ;
7      sh:prefixes inst:prefixes;
8      sh:select """
9       SELECT $this (?guid as ?value){
10       $this   ifc:globalId_IfcRoot/express:hasString ?guid .
11       FILTER NOT EXISTS { ?b props:globalIdIfcRoot_attribute_simple ?guid }
12       }
13     """
14     ]
```

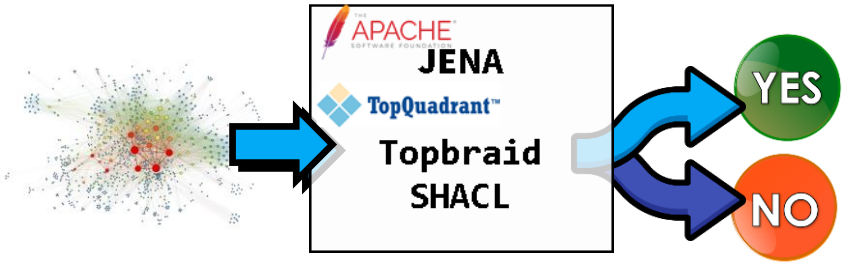
SHACL is a general-purpose data validation language and specification from the World Wide Web Consortium (W3C)

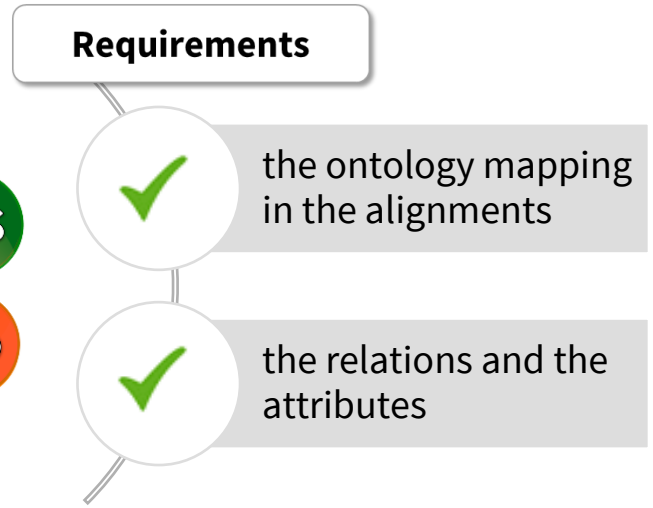


Wikimedia Commons: Lilyu

For selected subset of alignment:

- one-to-one correspondence (bijection):
  $x \mapsto f(y), y \mapsto f(x)$
- based on GUID



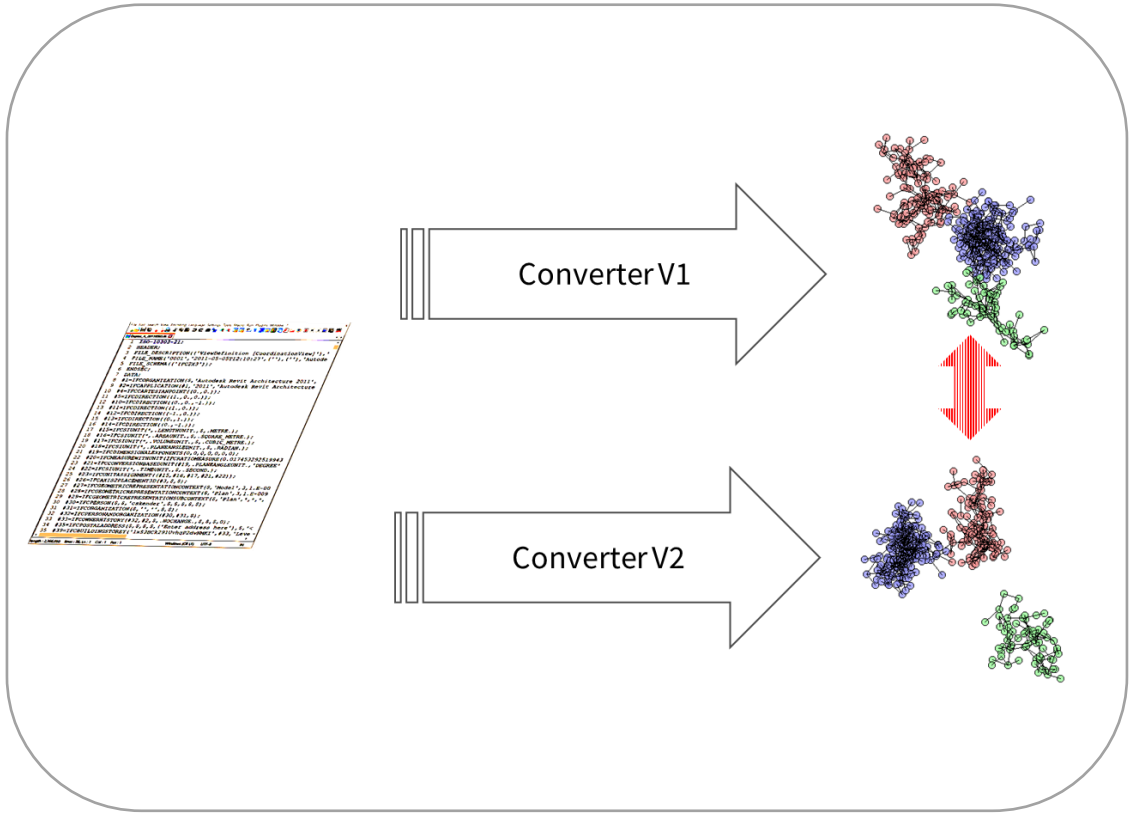


Yes/NO Image by Oberholster Venita from Pixabay
Split arrow: frlaticon

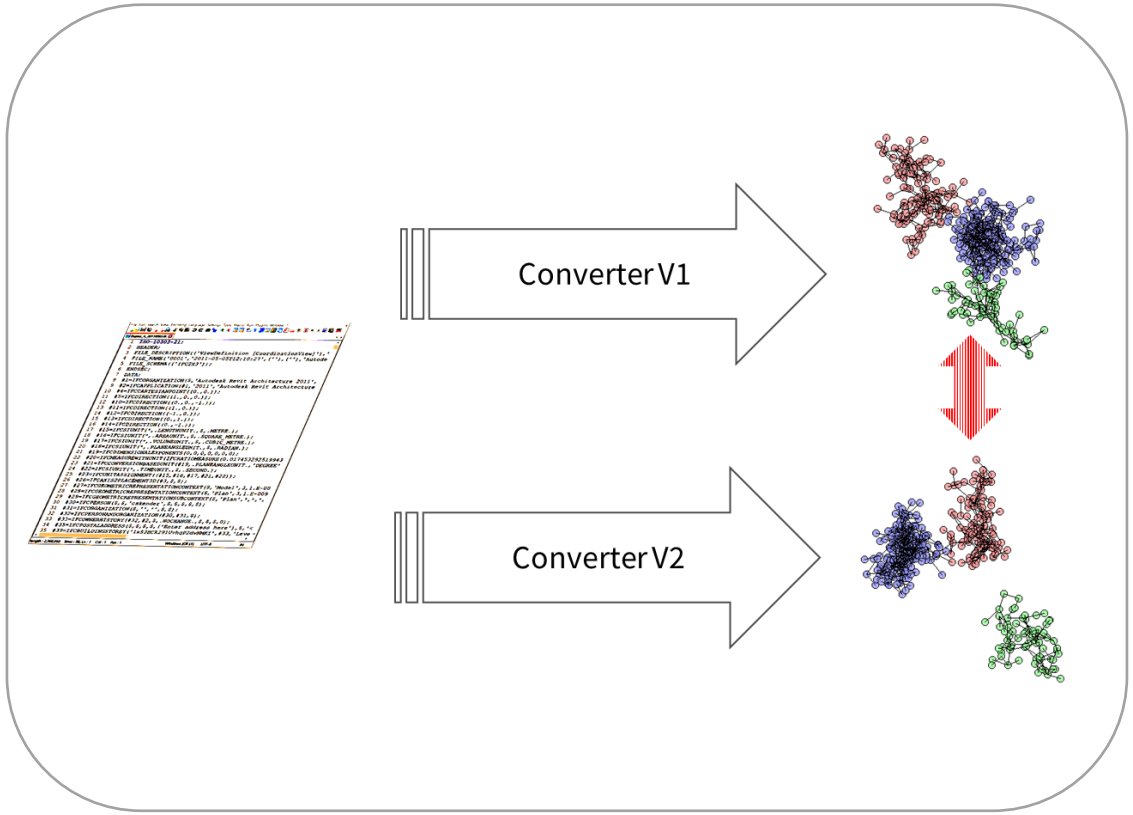
BIM4Ren

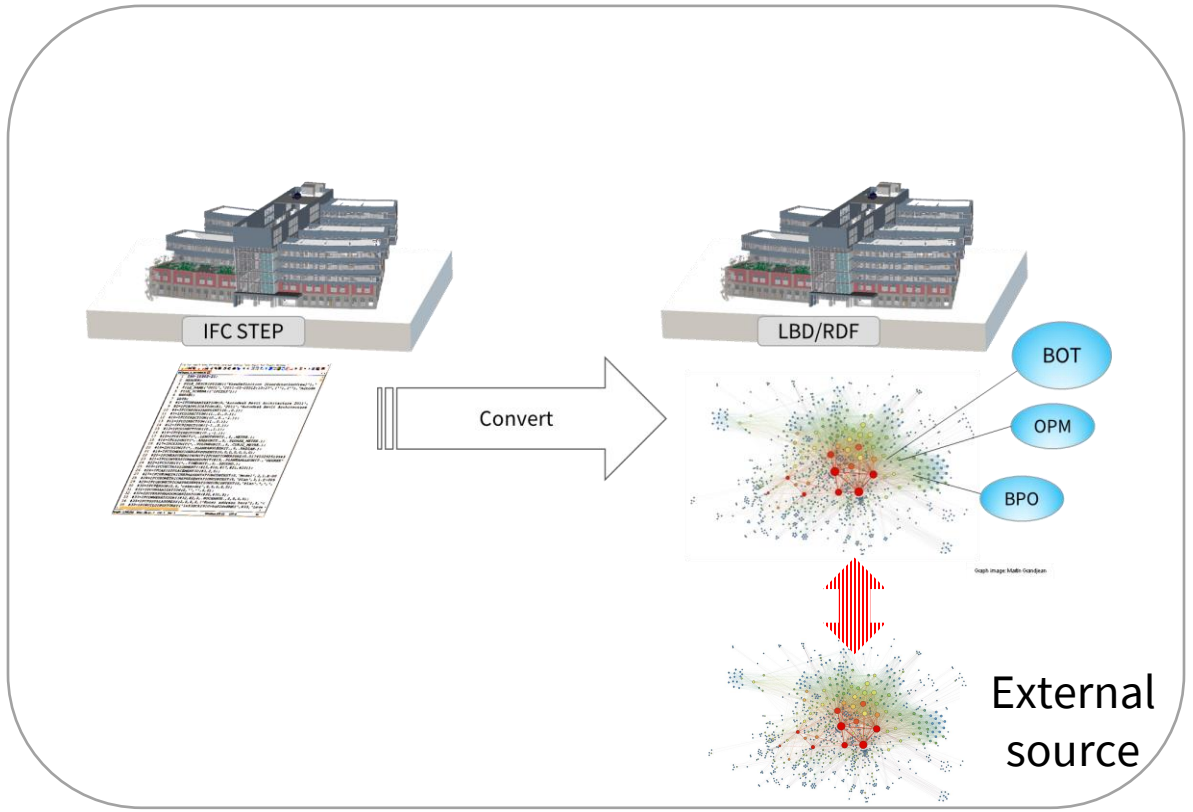
Design Computation | RWTH AACHEN UNIVERSITY

Unit testing: Compare with existing outputs

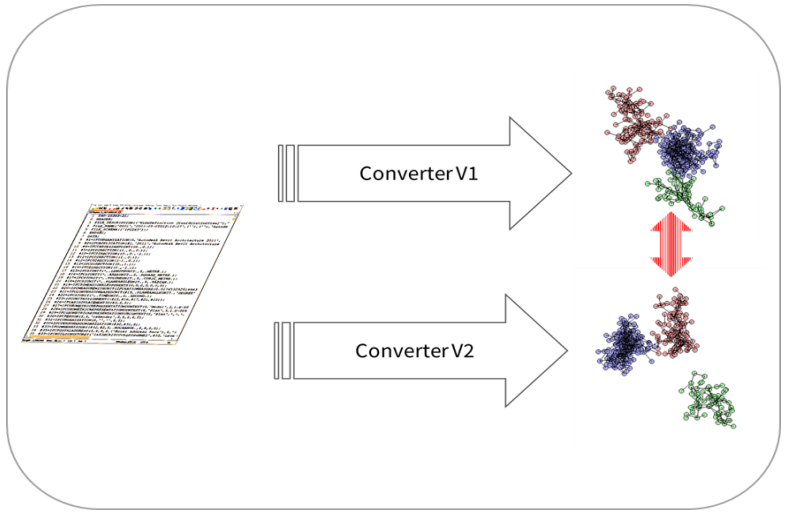


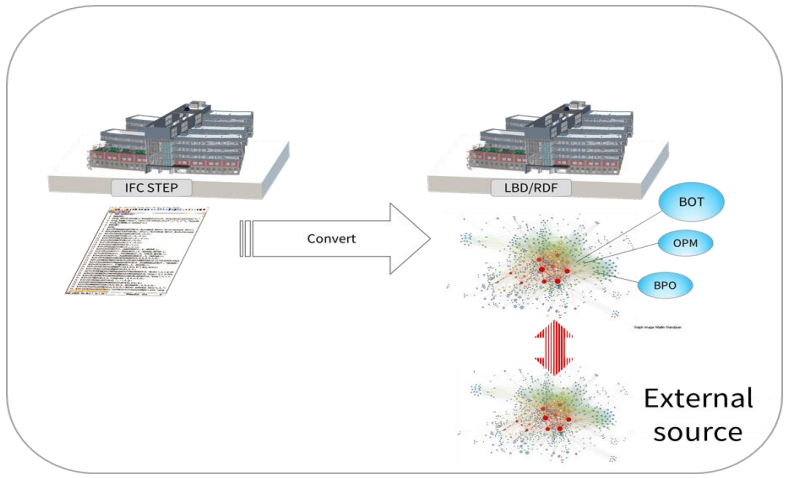
Unit testing: Compare with existing outputs





Unit testing: Compare with existing outputs

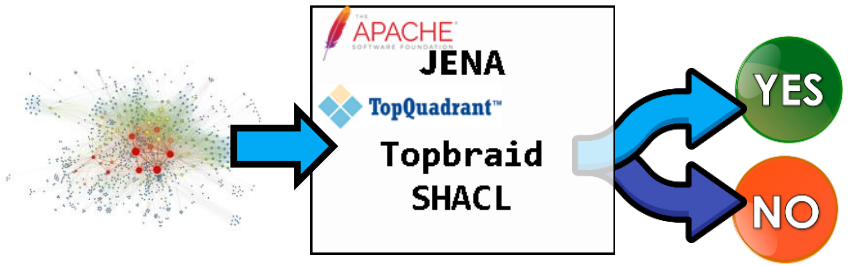




Listing 3. SHACL rule to check checksum of properties

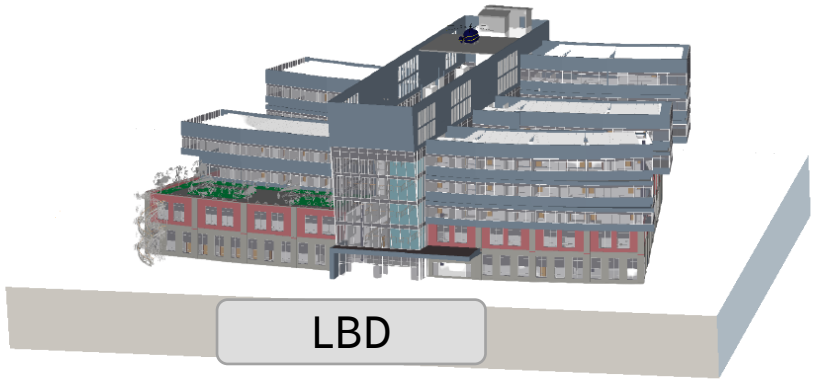
```
inst:shapeBOTChecksum a sh:NodeShape ;
  sh:targetNode inst:stairflight_ca457005-aa0d-4679-92c1-5067d702c9f3;
  sh:property [
  sh:sparql [
     a sh:SPARQLConstraint ;
     sh:message "The checksum of the properties is not valid." ;
     sh:prefixes inst:prefixes;
     sh:select """
       SELECT $this ?ResultSetHash WHERE {
       {
       SELECT $this (MD5(GROUP_CONCAT(CONCAT(STR(?p)); separator=' ')) as ?hash )
       WHERE  {
          SELECT *
          WHERE {
             $this ?p ?o.
        ORDER BY ?s ?p ?o
       } GROUP BY $this
      }
      FILTER (?hash != "82c7dc90fcb57319f2bb7ead58ead1de")
    }
    """
] .
```

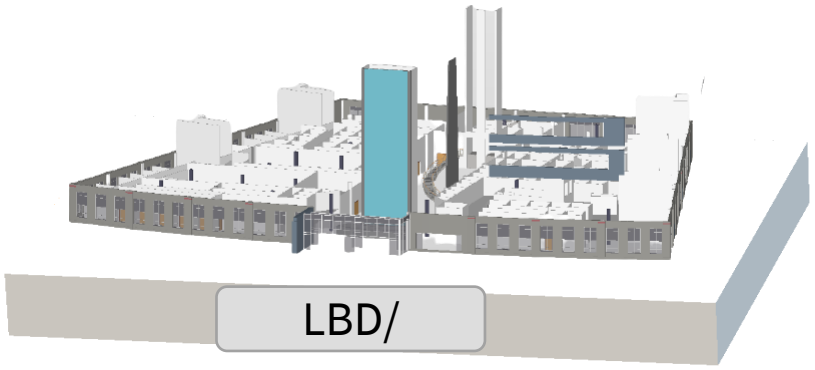
- Checksum-based solution
- Also, example-based SHACL generation

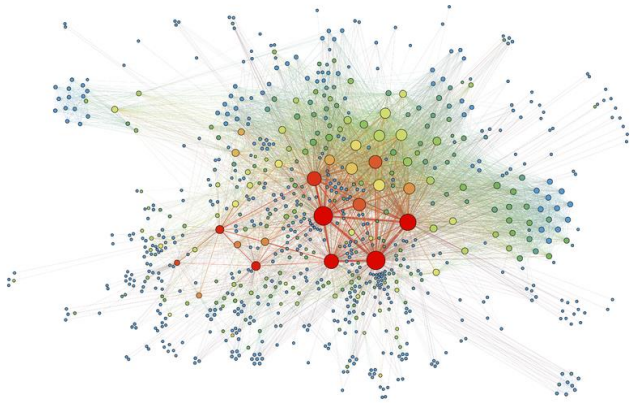


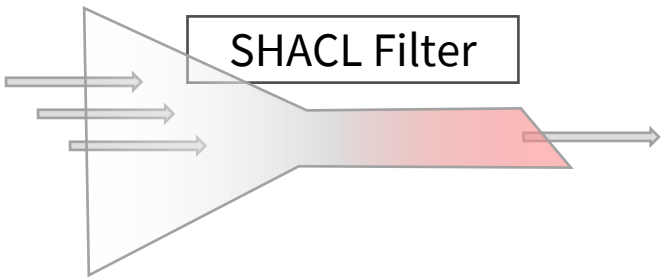
Model View Definition (MVD)

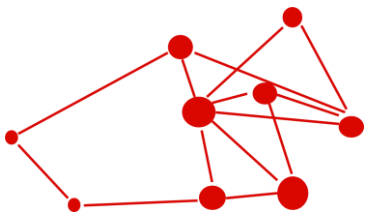
buildingSMART International has defined a **Model View Definition (MVD)** as a subset of IFC schema for a specific use case.





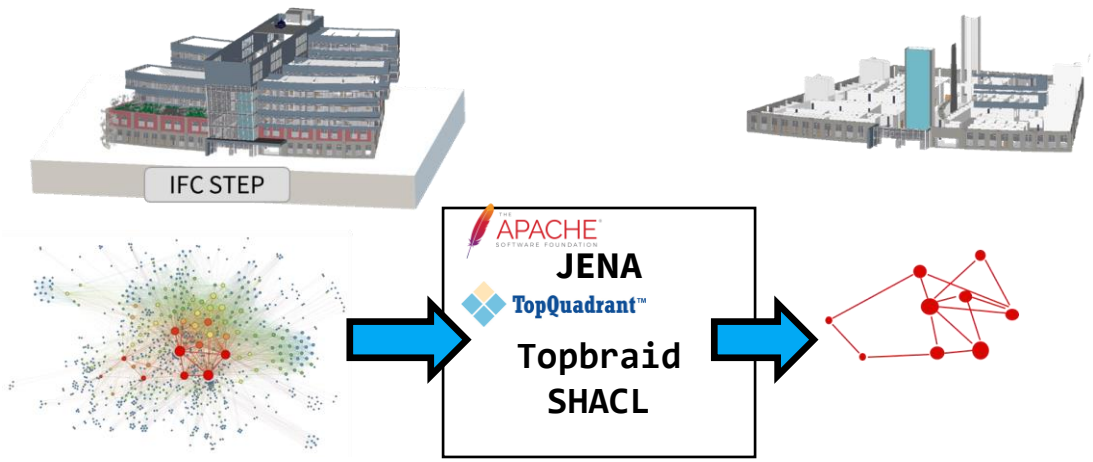






Graph image: Martin Grandjean

Model View Definition (MVD)



Listing 5. Sample SHACL Filter for Window elements

```
1    inst:HasGeometry
2        sh:path bot:hasSimple3DModel ;
3        sh:minCount 1 .
4
5    inst:RuleFilterWindowsWithGeometryShape
6      a sh:NodeShape ;
7      sh:targetClass bot:Element ;
8      sh:property inst:HasGeometry ;
9      sh:rule [
10       a sh:SPARQLRule;
11       rdfs:label "Construct a Geometry MVD";
12       sh:prefixes inst:prefixes ;
13       sh:construct """
14         CONSTRUCT {
15           $this ?p ?o .
16         }
17         WHERE {
18           $this ?p ?o .
19           FILTER (regex (STR($this),"window_")).
20         }
21         """
22     ] .
```

Conclusions

- So far, there have not been **unit tests** that focus on the ontology translation process of an LBD model.
- This study presents a couple of unit test methods to improve **conversion quality**.
- Besides validation, **model views** can be created for LBD data using SHACL

Future work

- More extensive analysis
- Standard model view creation.

Thank you for your attention!

BIM4Ren

BIM4Ren

Design Computation | RWTH AACHEN UNIVERSITY

Data Validation

mvdXML

SPARQL

SHACL

Open-Source Data Validation Approaches

mvdXML

standard

- IFCDoc - creates XML rulesets preloads all the IFC schema releases.
  - enables checking the existence of a value/entity/attribute
  - accuracy
  - Cardinality
- complex nature of IFC
- only considers IFC schema
- MVD-based view constructors are not flexible and dynamic

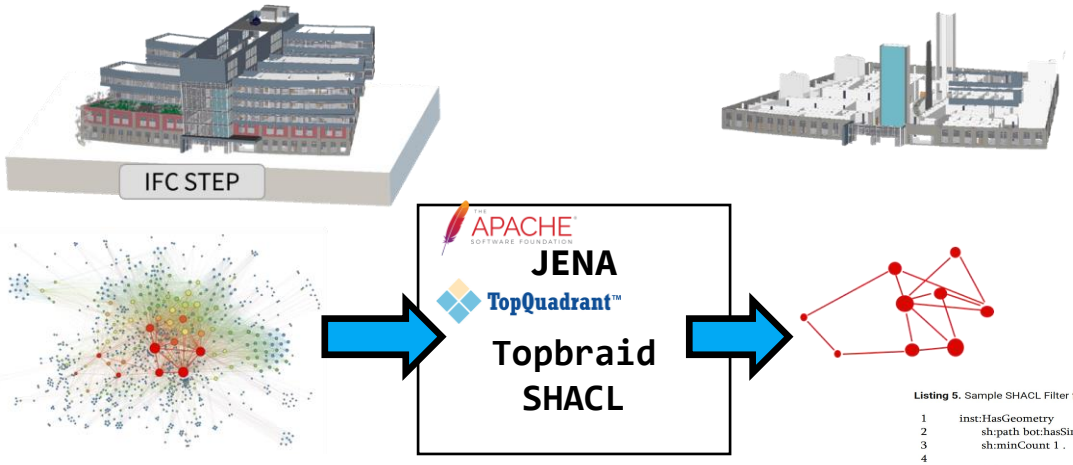
domain end-user to have knowledge of IFC, mvdXML and *IFCDoc*

SPARQL

W3C® recommendation

- verbosity and flexibility to define a constraint in multiple ways
- high threshold for learning
- user to enter queries according to SPARQL syntax

Model View Definition (MVD)



Listing 5. Sample SHACL Filter for Window elements

```
inst:HasGeometry
        sh:path bot:hasSimple3DModel ;
        sh:minCount 1 .

inst:RuleFilterWindowsWithGeometryShape
    a sh:NodeShape ;
    sh:targetClass bot:Element ;
    sh:property inst:HasGeometry ;
    sh:rule [
      a sh:SPARQLRule;
      rdfs:label "Construct a Geometry MVD";
      sh:prefixes inst:prefixes ;
      sh:construct """
          CONSTRUCT {
            $this ?p ?o .
          }
          WHERE {
            $this ?p ?o .
            FILTER (regex (STR($this),"window_")).
          }
      """
    ] .
```

```
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .

<https://ldc.de/window_6f010293-ddb7-4fe4-8ffd-c144d8444cde>
        rdf:type    <https://w3id.org/bot#Element> , <https://pi.pauwel.be/voc/buildingelement#Window> ;
        <http://lbd.arch.rwth-aachen.de/props#batid_attribute_simple>
                "181096" ;
        <http://lbd.arch.rwth-aachen.de/props#globalIdIfcRoot_attribute_simple>
                "1l0GAJtRTFv8$zmKJOH4pU" ;
        <http://lbd.arch.rwth-aachen.de/props#nameIfcRoot_attribute_simple>
                "M_Fixed:2800mm x 2410mm:2800mm x 2410mm:181096" ;
        <http://lbd.arch.rwth-aachen.de/props#objectTypeIfcObject_attribute_simple>
                "2800mm x 2410mm" ;
        <http://lbd.arch.rwth-aachen.de/props#overallHeightIfcWindow_attribute_simple>
                "2.409999999999996"^^xsd:double ;
        <http://lbd.arch.rwth-aachen.de/props#overallWidthIfcWindow_attribute_simple>
                "2.8"^^xsd:double ;
        <http://www.opengis.net/ont/geosparql#hasGeometry>
                [] ;
        owl:sameAs  <https://ldc.de/IfcWindow_22123> ;
        <https://linkebuildingdata.org/LBD#containsInBoundingBox>
                <https://ldc.de/space_0b74b3fa-1a92-405e-9ac9-d59067be1d66> .

<https://ldc.de/window_6b61ce71-1a7a-473c-8f87-4262e0bdcfa0>
        rdf:type    <https://pi.pauwel.be/voc/buildingelement#Window> , <https://w3id.org/bot#Element> ;
        <http://lbd.arch.rwth-aachen.de/props#batid_attribute_simple>
                "149278" ;
        <http://lbd.arch.rwth-aachen.de/props#globalIdIfcRoot_attribute_simple>
                "1hOSvn6df7F8_7GcBWlS_W" ;
        <http://lbd.arch.rwth-aachen.de/props#nameIfcRoot_attribute_simple>
                "M_Fixed:2800mm x 2410mm:2800mm x 2410mm:149278" ;
        <http://lbd.arch.rwth-aachen.de/props#objectTypeIfcObject_attribute_simple>
                "2800mm x 2410mm" ;
        <http://lbd.arch.rwth-aachen.de/props#overallHeightIfcWindow_attribute_simple>
                "2.409999999999996"^^xsd:double ;
        <http://lbd.arch.rwth-aachen.de/props#overallWidthIfcWindow_attribute_simple>
                "2.8"^^xsd:double ;
        <http://www.opengis.net/ont/geosparql#hasGeometry>
                [] ;
        owl:sameAs  <https://ldc.de/IfcWindow_7740> ;
        <https://linkebuildingdata.org/LBD#containsInBoundingBox>
                <https://ldc.de/beam_98d604b7-4c6f-4ae8-b5f3-0187bb53f31b> .
```