Using BIM for Energy-Efficient Renovation

Pierre Bourreau – Nobatek/INEF4

Nathalie Charbel (Nobatek/INEF4) – Madhumitha Senthilvel (RWTH) – Jakob Beetz (RWTH) – Nicolas Bus (CSTB) – Michel Böhms (TNO) – Jyrki Oraskari (RWTH)...

This project has received funding from the H2020 programme under Grant Agreement No. 820773

Outline

1. BIM and Renovation
   - What are the challenges
   - BIM4Ren structure
   - Experiment as much web semantic technologies as possible.
2. The BIM4Ren data model
   - Need to model uncertainty – OPM
   - Need to model building status
   - Data access – SPARQL/GraphQL-LD/RAMOSE
3. The conversion process
   - Based on SWRL rules
4. Additional features
   - SHACL checkers
   - A product of catalogue

BIM4Ren overview

BIM in Renovation

- BIM is mainly used on new projects
  - Need to digitize existing buildings (not just scanning)
  - Need a suitable model (IFC extension)…
  - … Usable in the construction & exploitation phases
- BIM and Renovation: a complementarity
  - Can be combined with Energy Audit interventions
- Different depths of renovation projects
  - Different Information for different projects

BIM in Renovation

BIM4Ren

- BIM is mainly used on new projects
  - Need to digitize existing buildings (not just scanning)
  - **Need a suitable model (IFC extension)…**
  - … Usable in the construction & exploitation phases
- BIM and Renovation: a complementarity
  - Can be combined with Energy Audit interventions
- Different depths of renovation projects
  - Different Information for different projects

H2020 G.A. 820773

BIM4Ren Concept

BIM4Ren

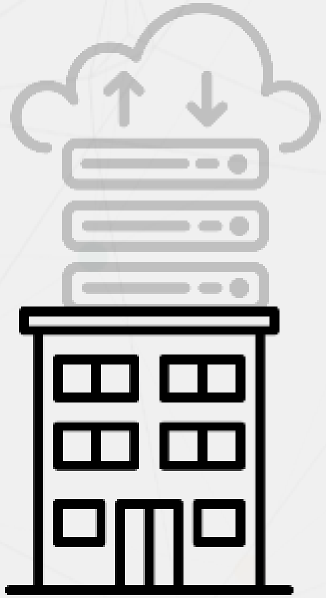
Data Management

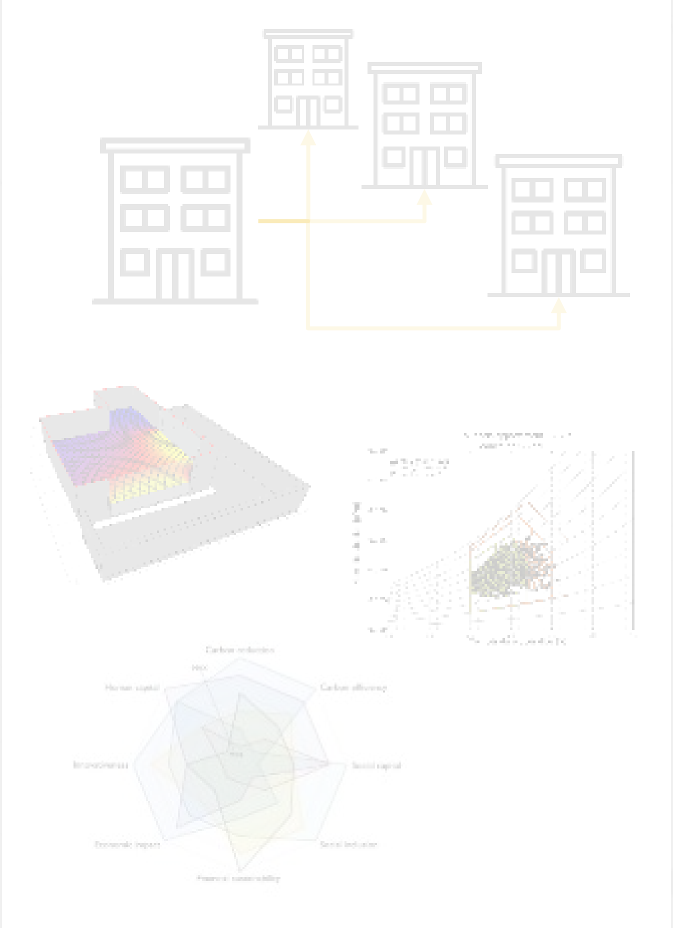
Data collection

Data- driven design

WHAT IS THE EXISTING DATA ?

- Year?
- Local regulation?
- Cost € ?
- Energy performance?
- Geometry?
- Stakeholders expectations?
- Type of occupants ?
- Renovation potential ?
- State of the existing infrastructure ?





Stakeholders involved in the renovation

BIM4Ren Concept

Digital inputs

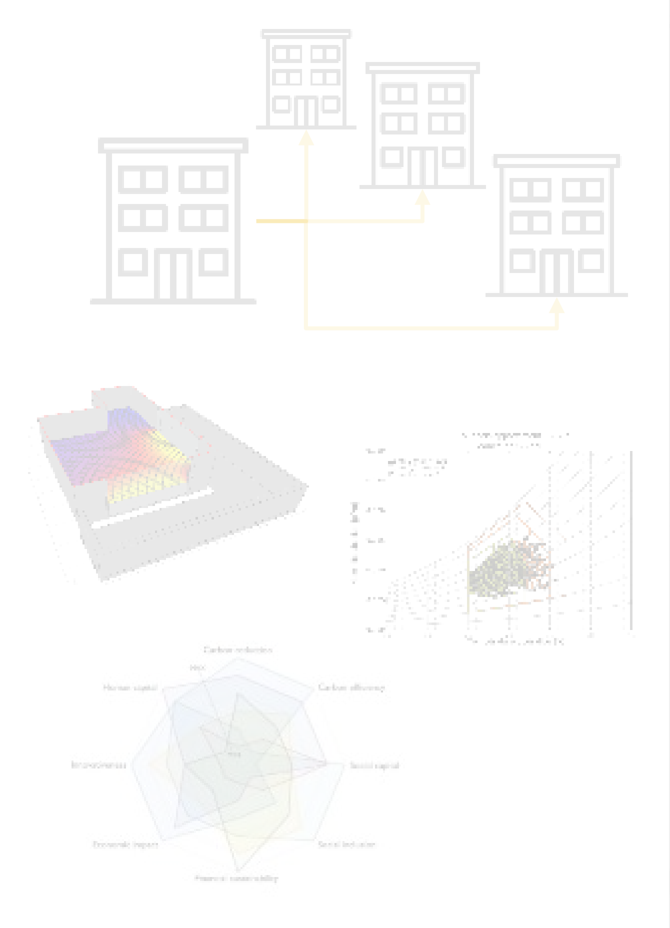
Data Management

Data collection

Data- driven design

WHAT IS THE EXISTING DATA ?

1. To create the BIM model
2. To organize, consolidate, secure
3. To validate the BIM model



Stakeholders involved in the renovation

BIM4Ren Concept

Digital inputs

Data Management

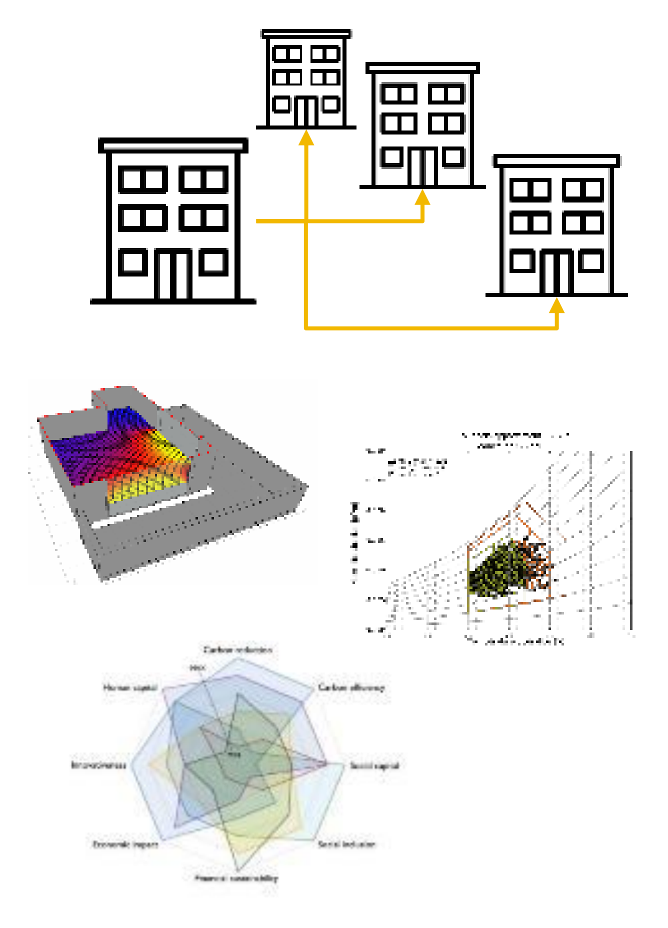
Exploit BIM

Data collection

Data- driven design

WHAT IS THE EXISTING DATA ?

1. To create the BIM model
2. To organize, consolidate, secure
3. To exploit the BIM model



Stakeholders involved in the renovation

BIM4Ren Concept

Digital inputs

Data Management

Data

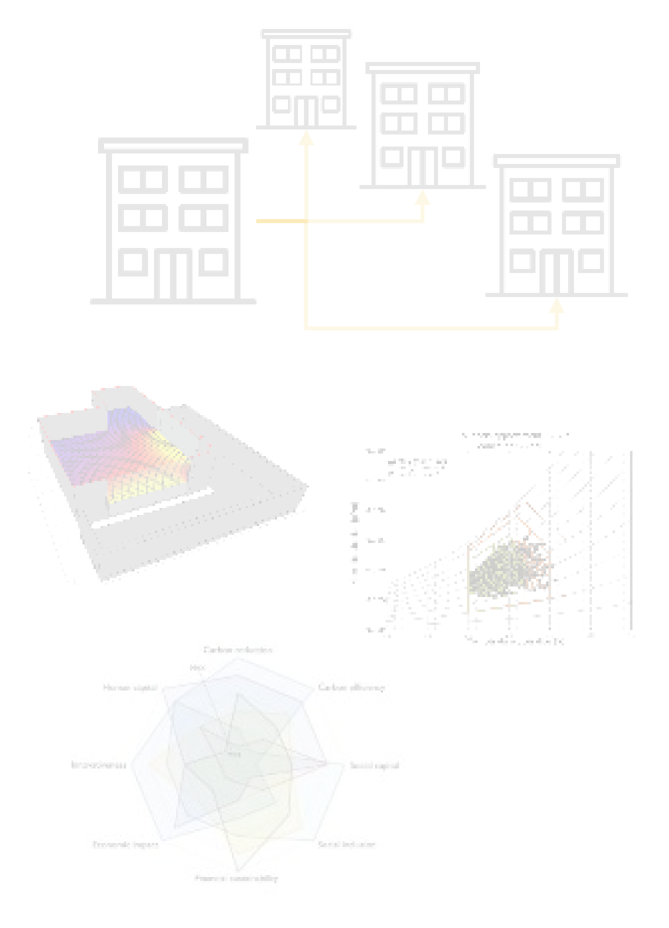
WHAT IS T
EXISTING D

Web semantic tools sandbox

- triple store
- BIM model for renovation
- checkers
- converters
- data access

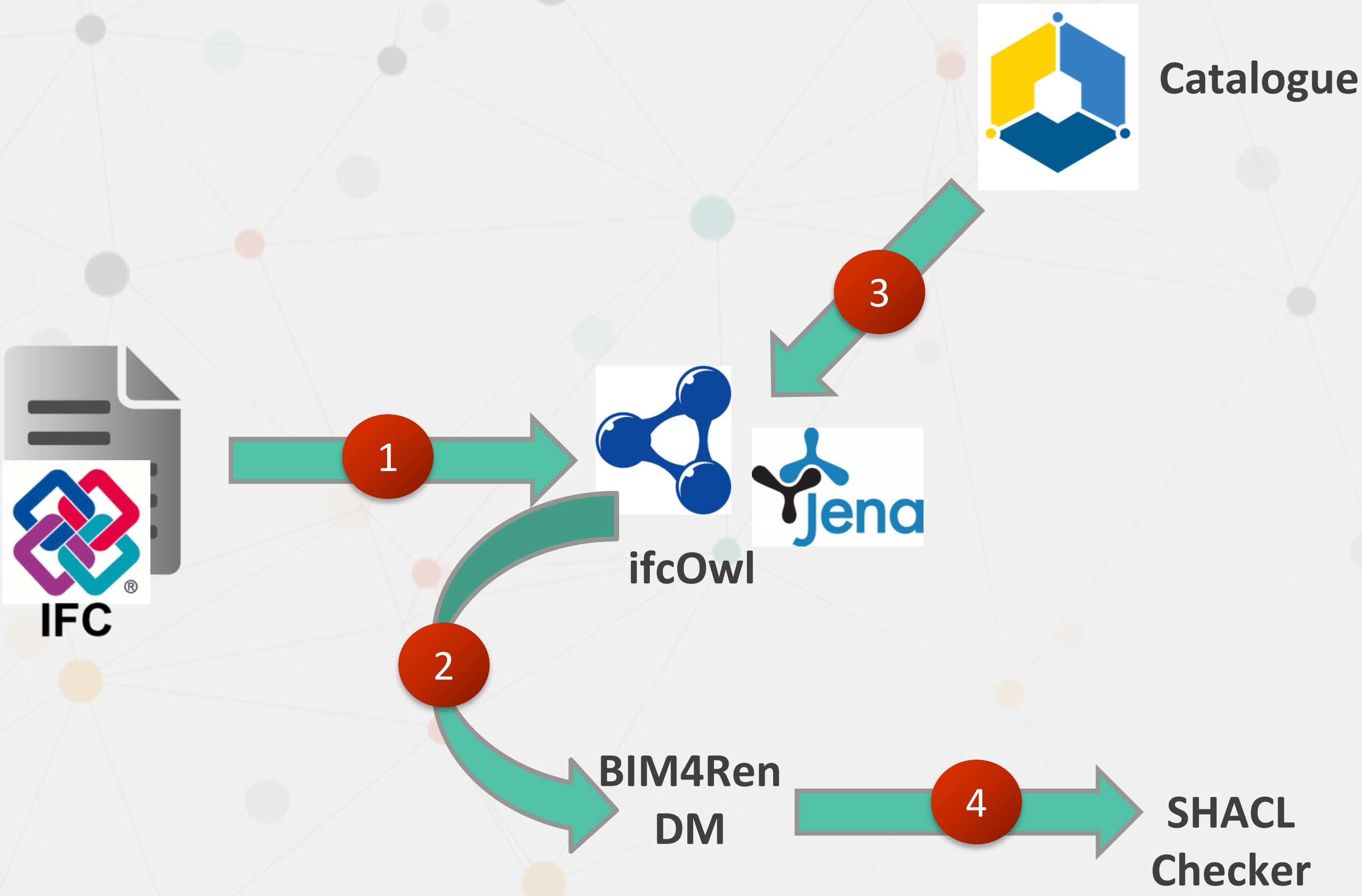
To create the BIM model
To organize, consolidate, secure
To validate the BIM model

Data- driven design



Stakeholders involved in the renovation

Deeper view



1 Conversion SPF to ifcOwl – loading to a triple store

2 Alignment with BIM4Ren data model

3 Enrichment from a LD catalogue

4 Checking information on models

The BIM4Ren data model

Goals

1 Need to model **uncertainty**

- ‘the composition of the external wall is **certainly brick + Xcm of glass wool**’
- The scan process has an **accuracy of X**

2 Need for **modularity**

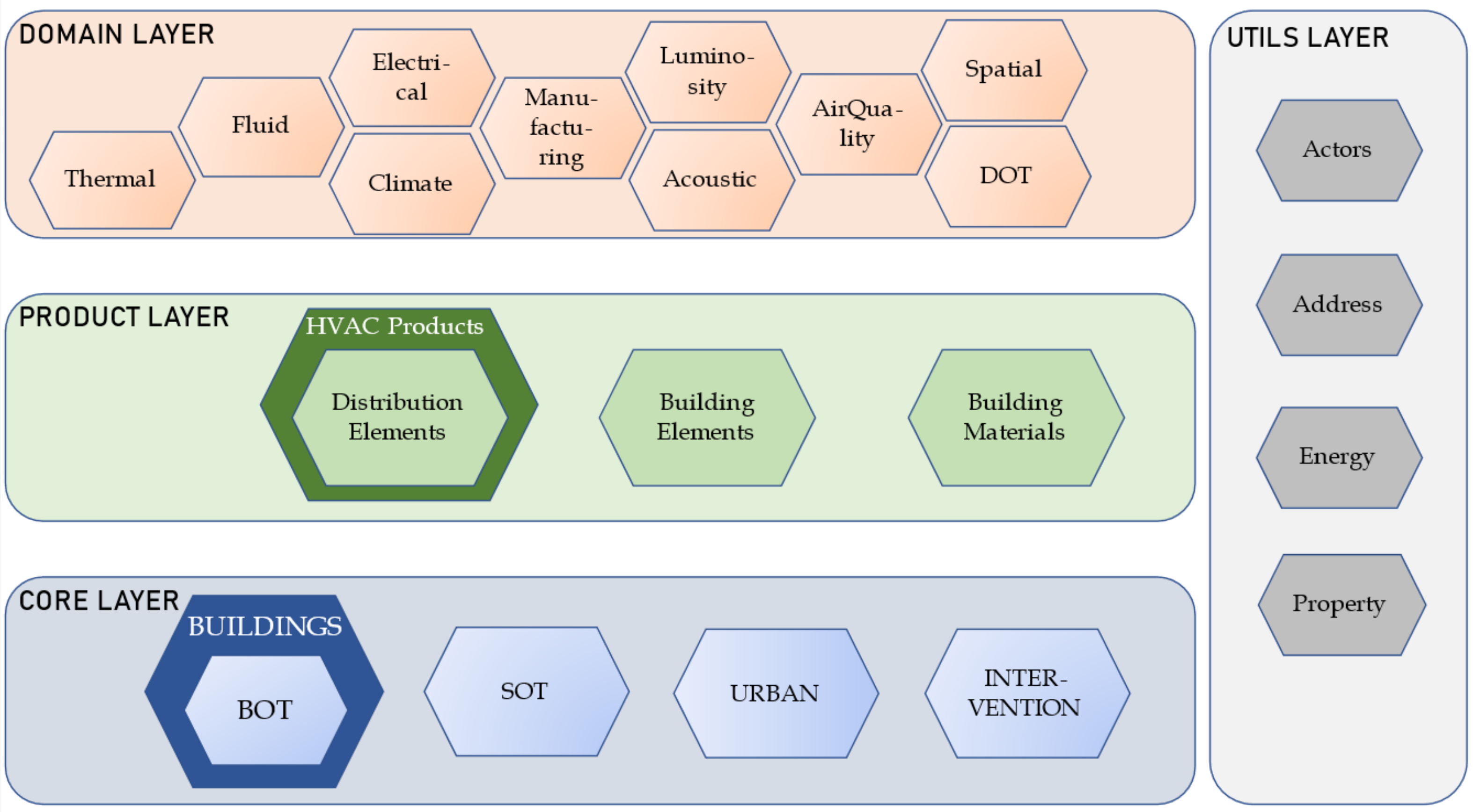
- Different type of renovation, different depth of renovation
- Models as simple as possible

Overview of the data model

https://models.bim4ren.eu

Modularity

- Independent ontologies
- 3 layers (quite similar to IFC layers)
- Design pattern through multiple inheritance



The Core layer

To describe specific aspects of a building, and to attach elements in it according to some specific properties

- **BOT**: Structure of the building
- **SOT**: Systems/network topology (MEP…)

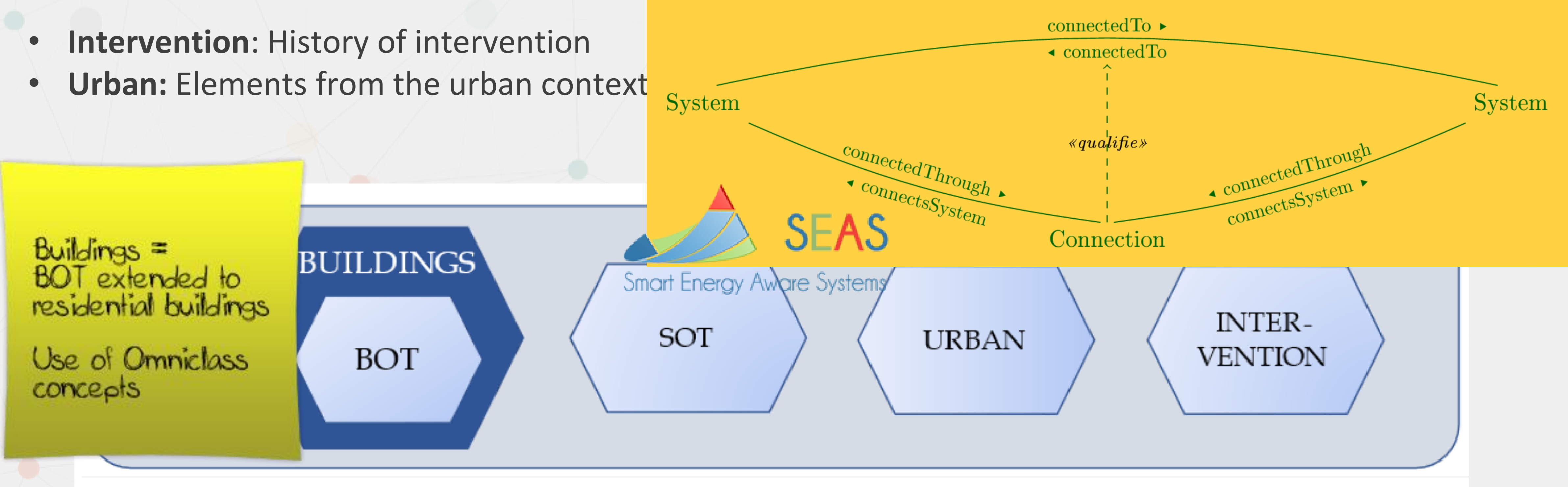
- **Intervention**: History of intervention
- **Urban:** Elements from the urban context

<u>Goal</u>: localize elements ('spatially')

<u>Goal</u>: what is the role of an element within the network? To which elements is it directly connected?

..)

..


Buildings = BOT extended to residential buildings

Use of Omniclass concepts

The Product layer

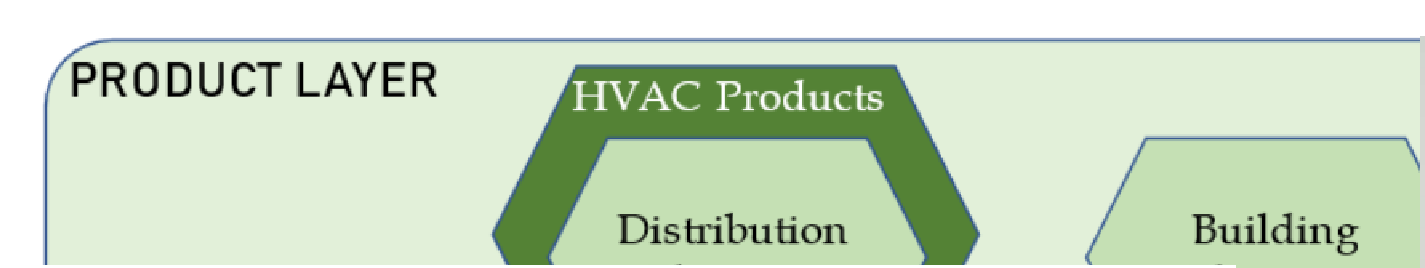
To categorize building products

- DistributionElements
- BuildingElements:
- BuildingMaterials

Goal: categories of products (distribution elements)

Goal: categories of products (building elements)

Goal: categories of materials (for LCA, BEM…)



Ontologies

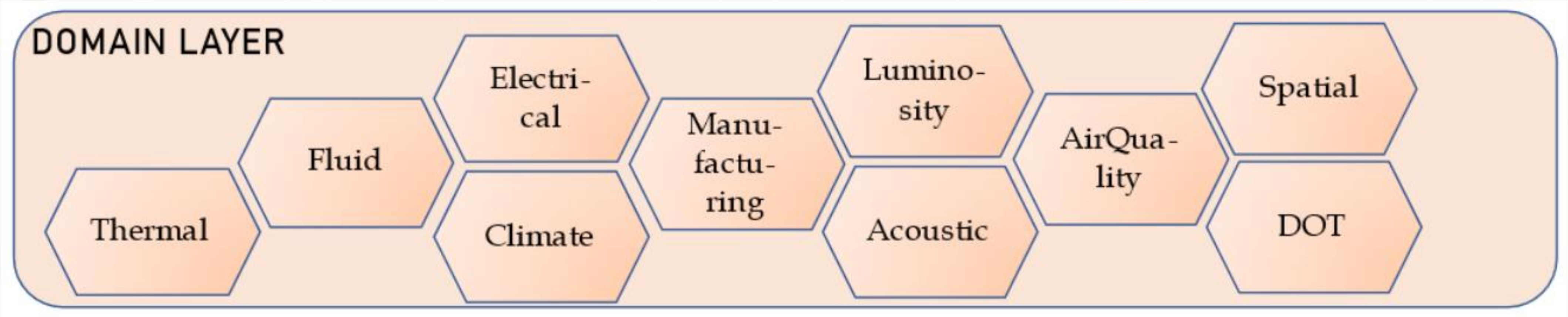
Product ontologies

Building Element: https://pi.pauwel.be/voc/buildingelement

Distribution Element: https://pi.pauwel.be/voc/distributionelement

Civil Element: https://pi.pauwel.be/voc/civilelement

The Domain layer

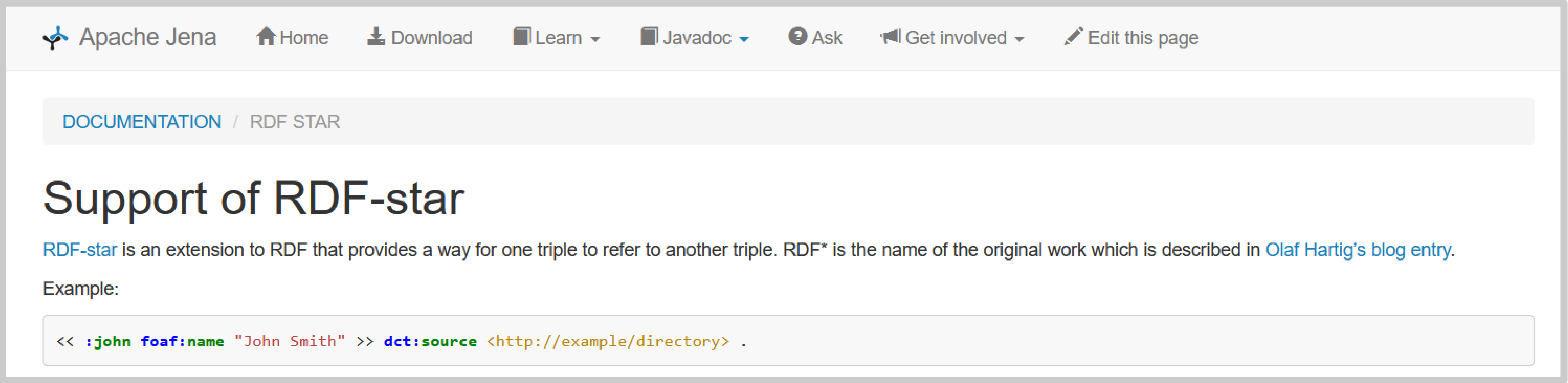


To associate specific domain properties to elements

- **Thermal**: thermal resistance, solar factor, coefficient of performance…
- **Fluid**: capacity, pressure…
- **Electrical**: nominal power, power in/out…
- **Manufacturing:** brand, product

…

The Property layer



DOCUMENTATION / RDF STAR

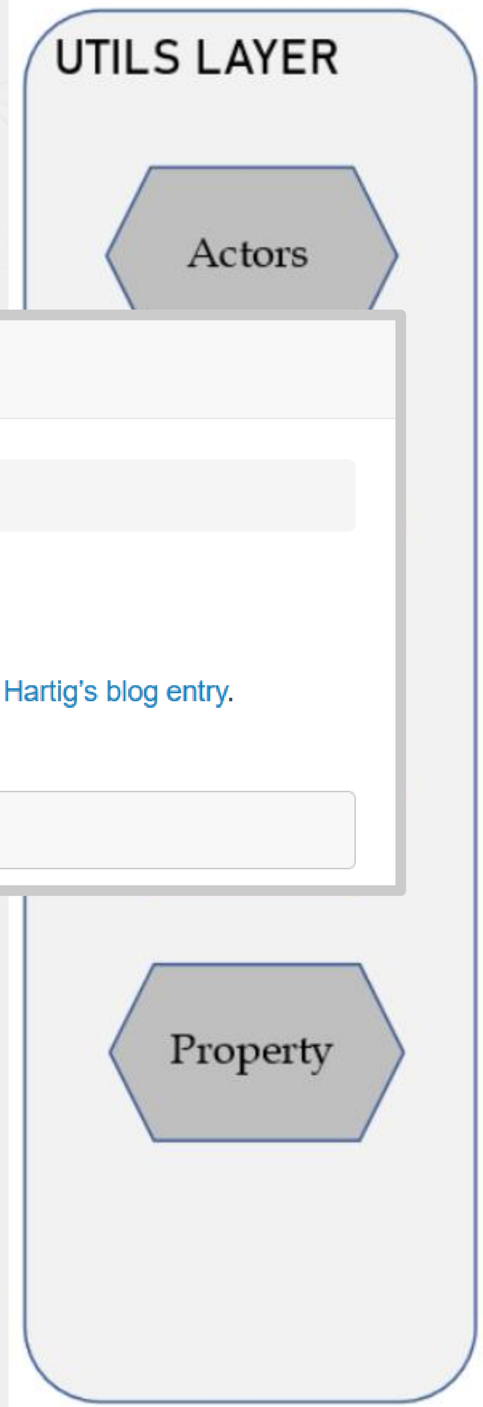
Support of RDF-star

RDF-star is an extension to RDF that provides a way for one triple to refer to another triple. RDF* is the name of the original work which is described in Olaf Hartig's blog entry.

Example:

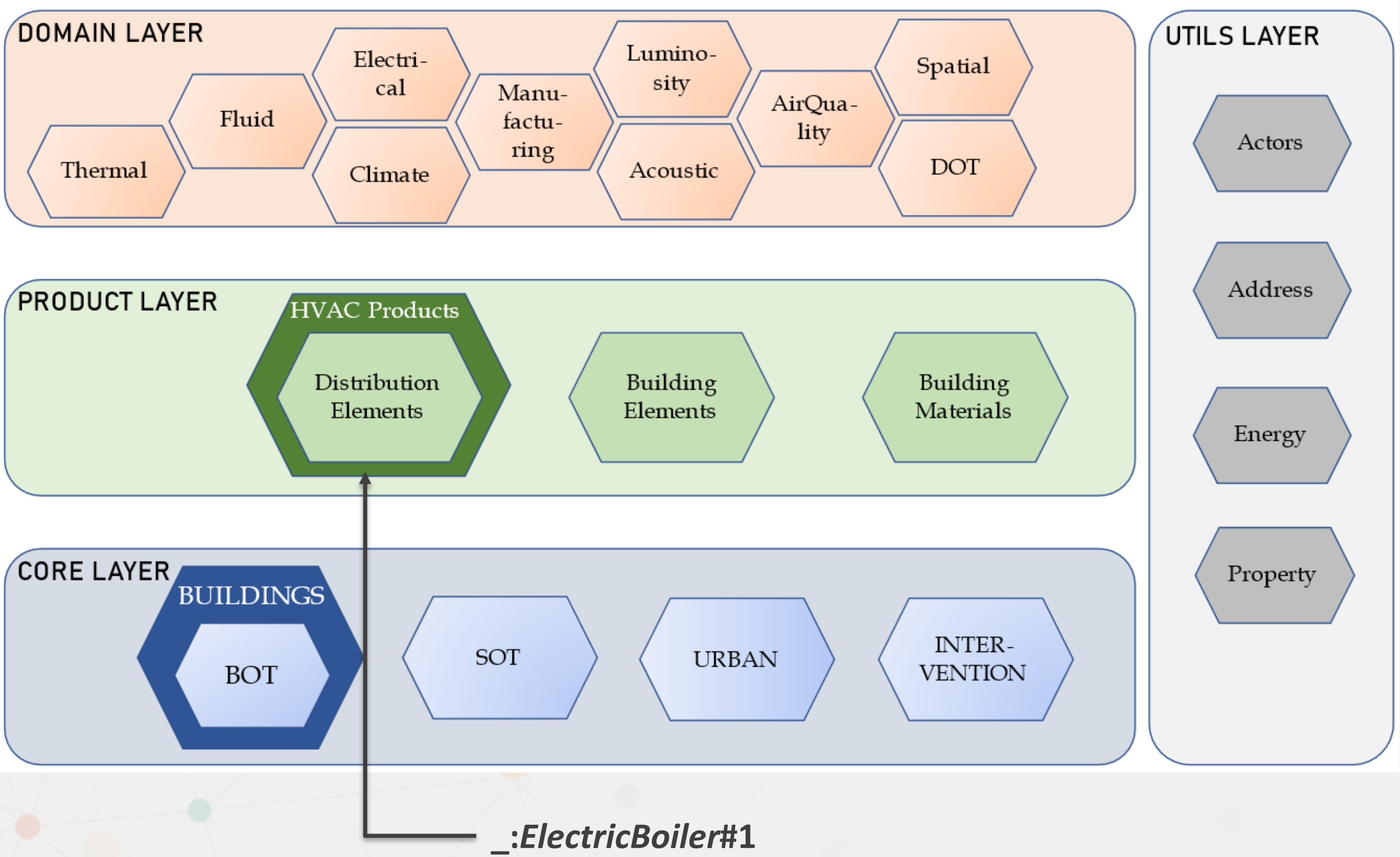
```
<< :john foaf:name "John Smith" >> dct:source <http://example/directory> .
```

- Next candidate: RDF* !!!



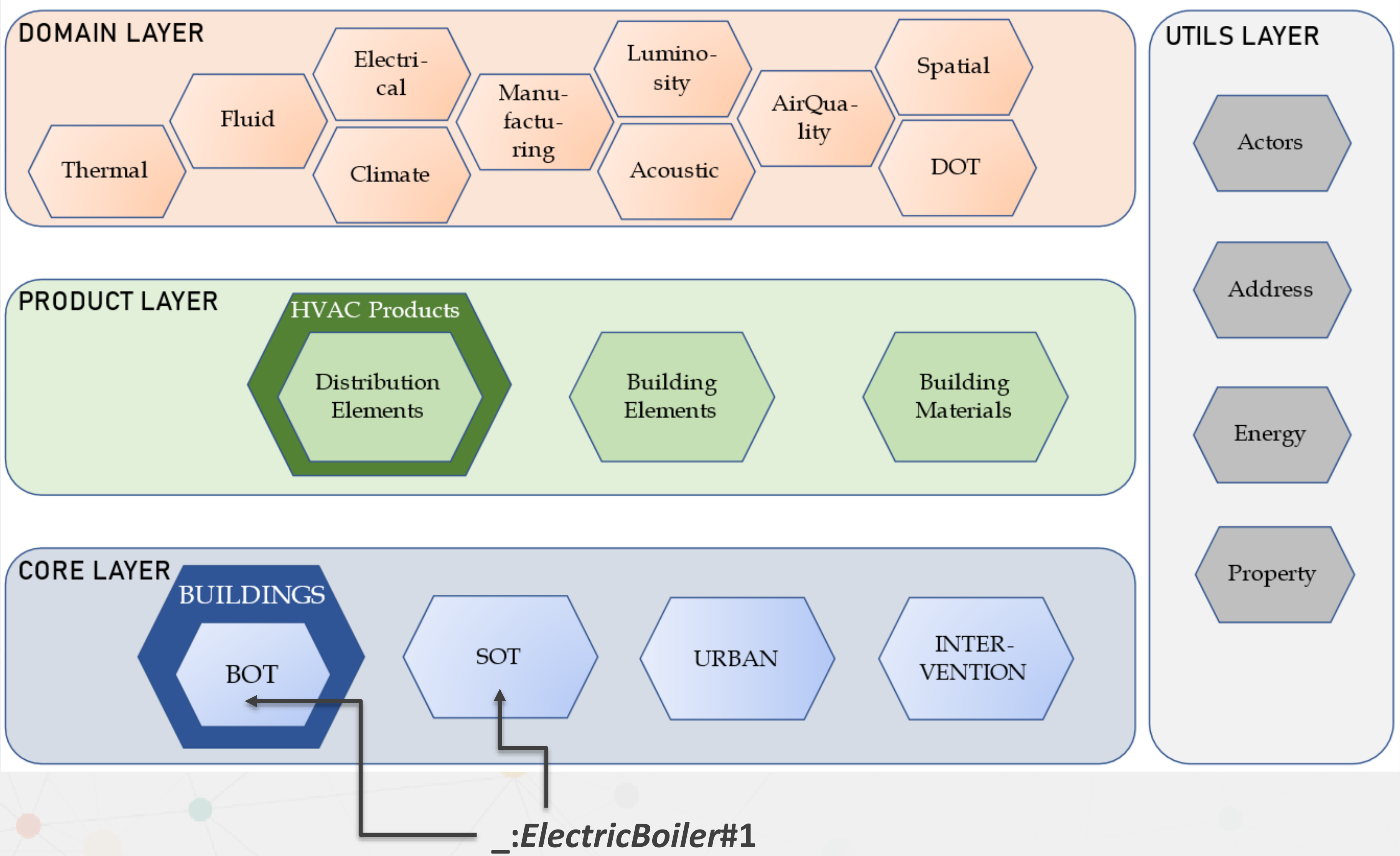
H2020 G.A. 820773

Full Modularity



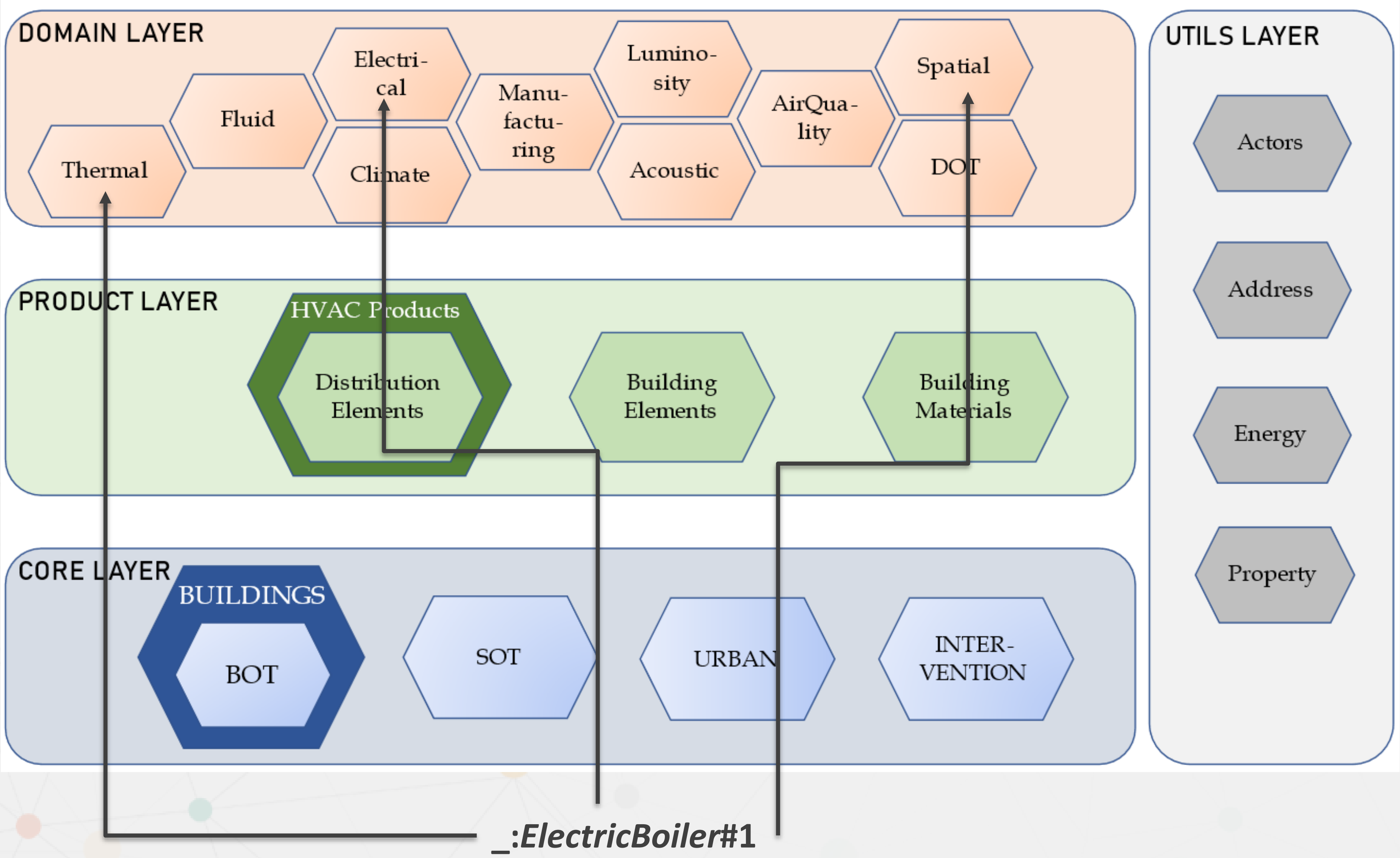
1. What type of product is this?

Full Modularity



1. What type of product is this?
2. Where is it located?
   - In the building
   - In my heating network.

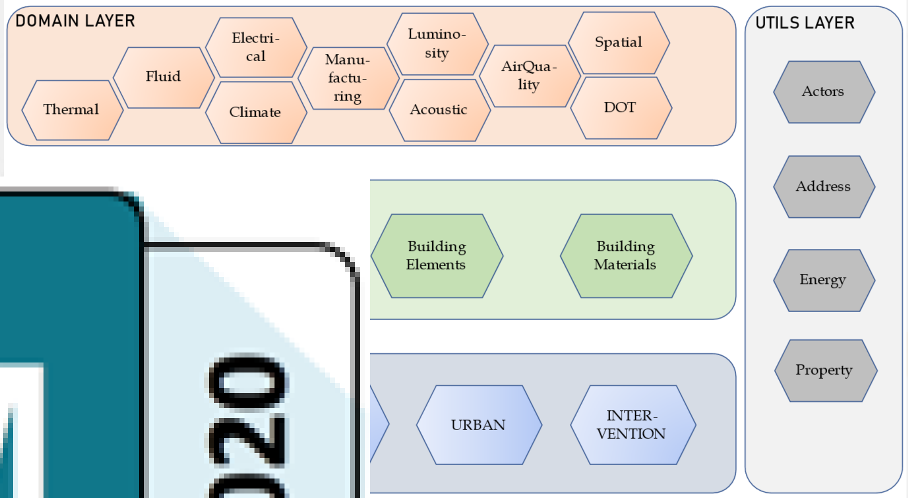
Full Modularity



1. What type of product is this?
2. Where is it located?
   - In the building
   - In my heating network.
3. What properties does it have?
   - For a thermal study
   - For an electrical study/dimensioning
   - Geometrical

Overview of the data model

- Modularity is being discussed in bSI
- Can a simple and modelling language-wide mechanism solv
- The model is **pr**
  - Product layer t
  - Core layer to as
  - Domain layer t Sets…)
- But fr
- Need under

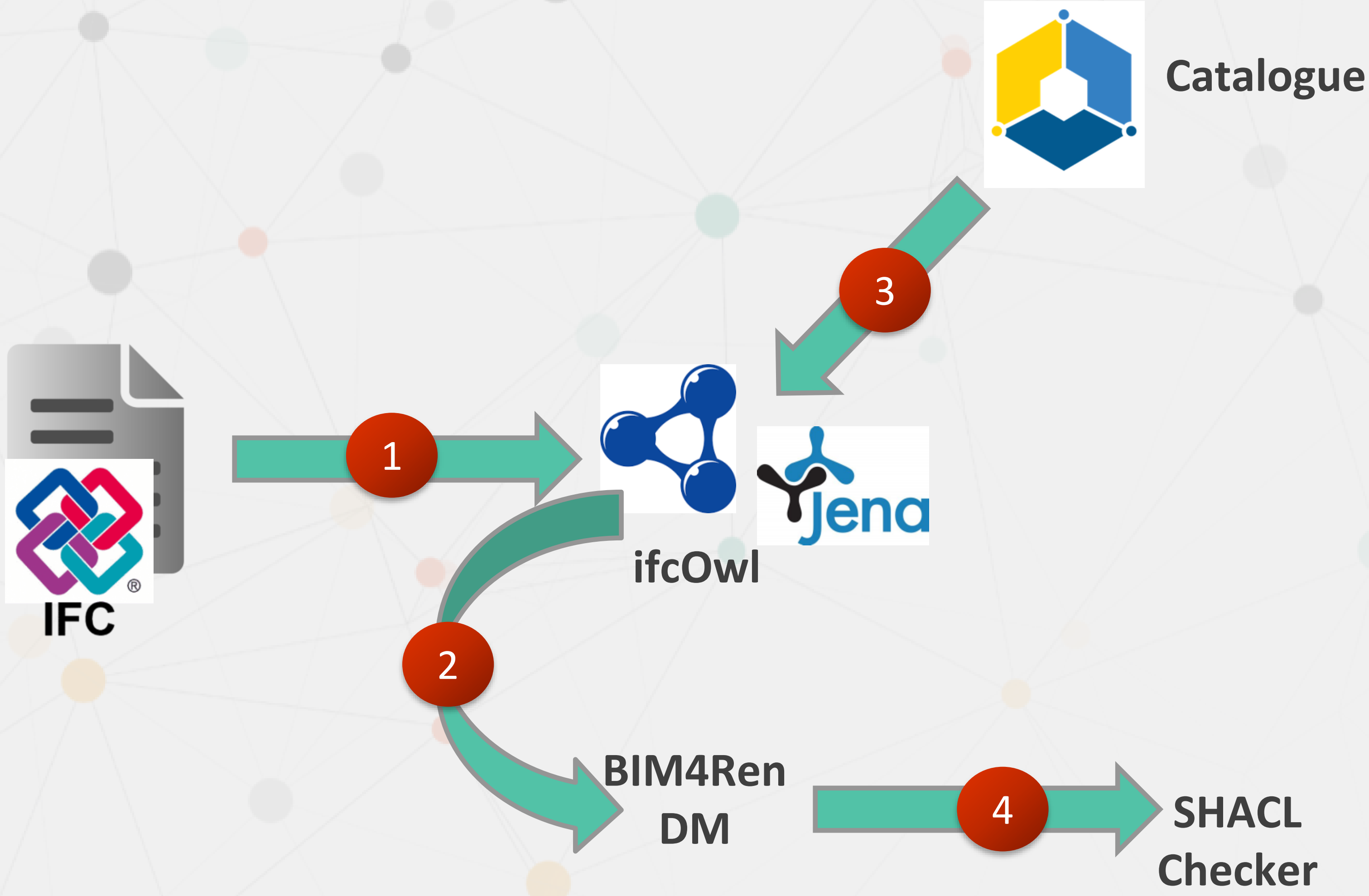


Multiple inheritance for a modular BIM

Pierre Bourreau[1], Nathalie Charbel[1], Jeroen Werbrouck[2], Madhumitha Senthilvel[3], Pieter Pauwels[4] and Jakob Beetz[3]

The conversion process

Why do we need conversion?



1 Conversion SPF to ifcOwl – loading to a triple store

2 Alignment with BIM4Ren data model

3 Enrichment from a LD catalogue

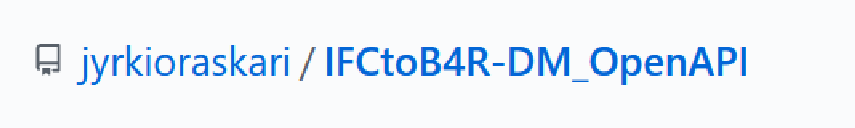
4 Checking information on models

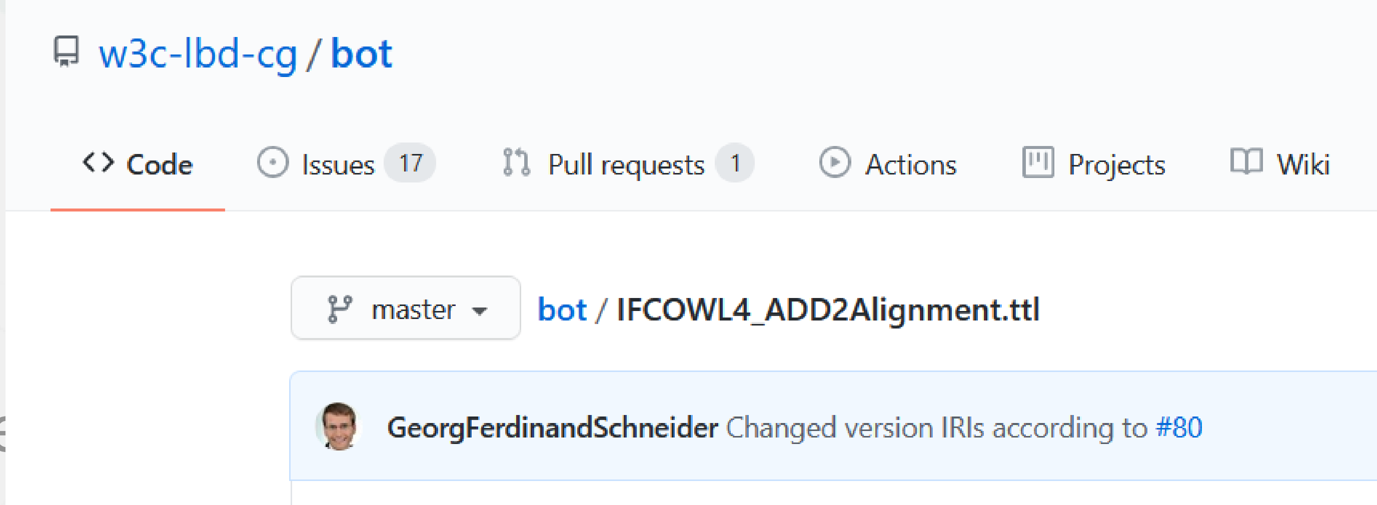
Conversion

1. IFC is the most common and open standard
   - We get IFCs from WiseBIM, ARtoBuild (geometry)
2. But IFC is quite complex => BIM4Ren model
   - Relations are objectified
   - Not all properties we want…
3. Enrich a geometric model with semantic information
   - CSTB develops POBIM based on ISO23386 - Ontology
   - BIM4Ren uses SML (CEN to-be standard)
4. CSTB: POBIM catalogue + ifcOwl
   - **Need to propagate changes on the BIM4Ren DM**
   - **ifcOwl to B4R (BOT+)** and **POBIM to SML**

Different options

- Static/programmatic conversion
  - ++Can cover all types of elements (geometry, relations...)
  - --Need save/convert/import procedure on changes
- Alignment
  - ++Dynamic in the triple store
  - --1-1 mapping
  - --Mostly covers classes, not relations


jyrkioraskari / IFCtoB4R-DM_OpenAPI

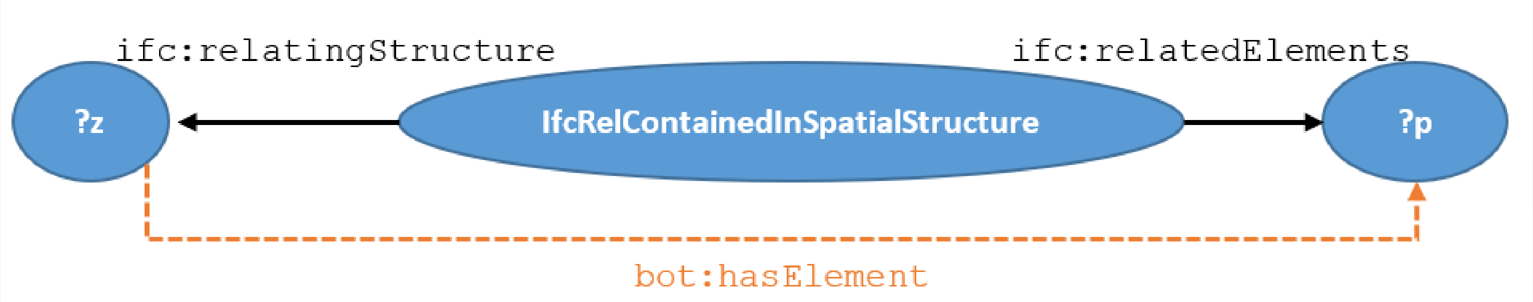

w3c-lbd-cg / bot

bot / IFCOWL4_ADD2Alignment.ttl

GeorgFerdinandSchneider Changed version IRIs according to #80

Relation mappings as Inference

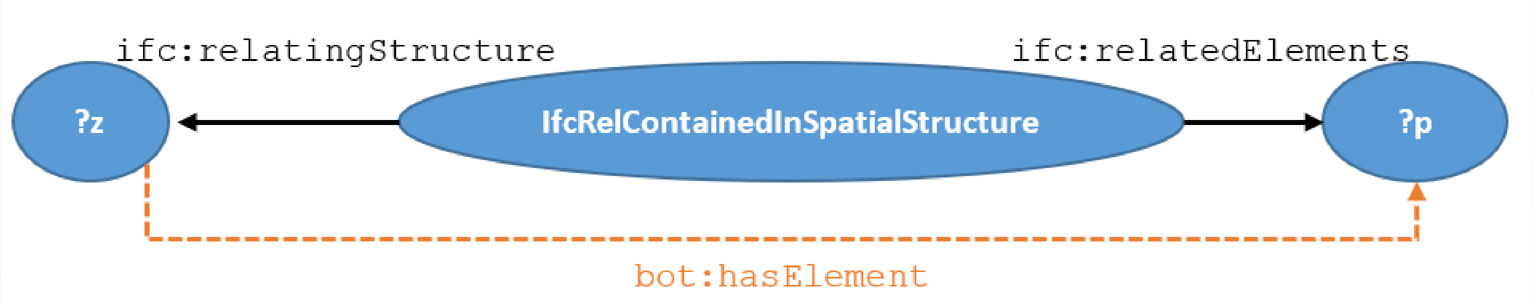
- IFC relations are objectified
- The mapping can be expressed in logic



If ?z is a bot:Zone
And ?z and ?c are connected by an
ifc:relatedElements_IfcRelContainedInSpatialStructure relation
=> Then <?z; bot:hasElement; ?p>

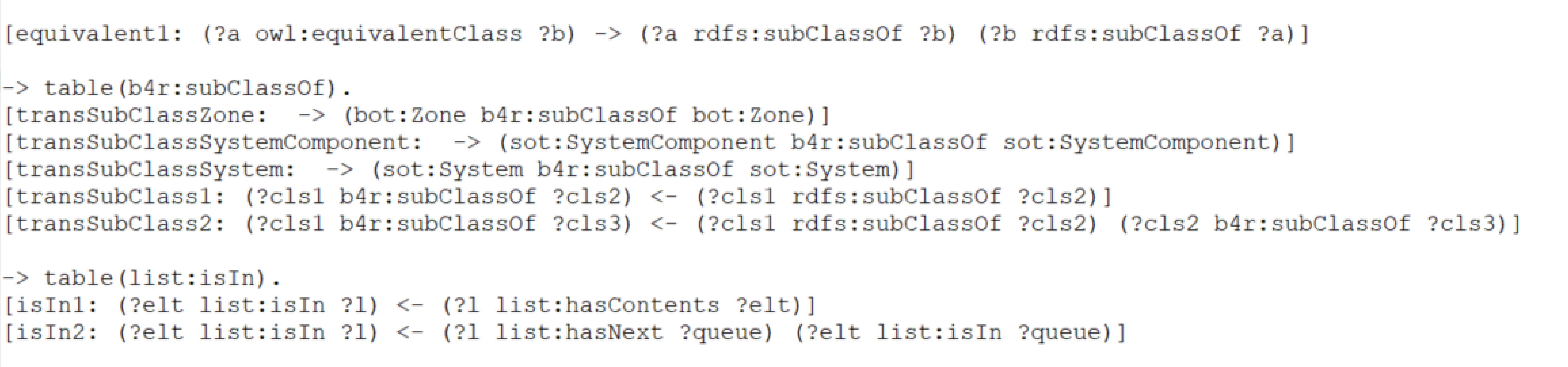
Relation mappings as Inference

- IFC relations are objectified
- The mapping can be expressed in logic



```
(?z rdf:type ?cl) (?cl rdfs:subClassOf* bot:Zone)
(?c ifc:relatingStructure_IfcRelContainedInSpatialStructure ?z)
(?c ifc:relatedElements_IfcRelContainedInSpatialStructure ?p)
=> (?z bot:hasElement ?p)
```

Utility rules

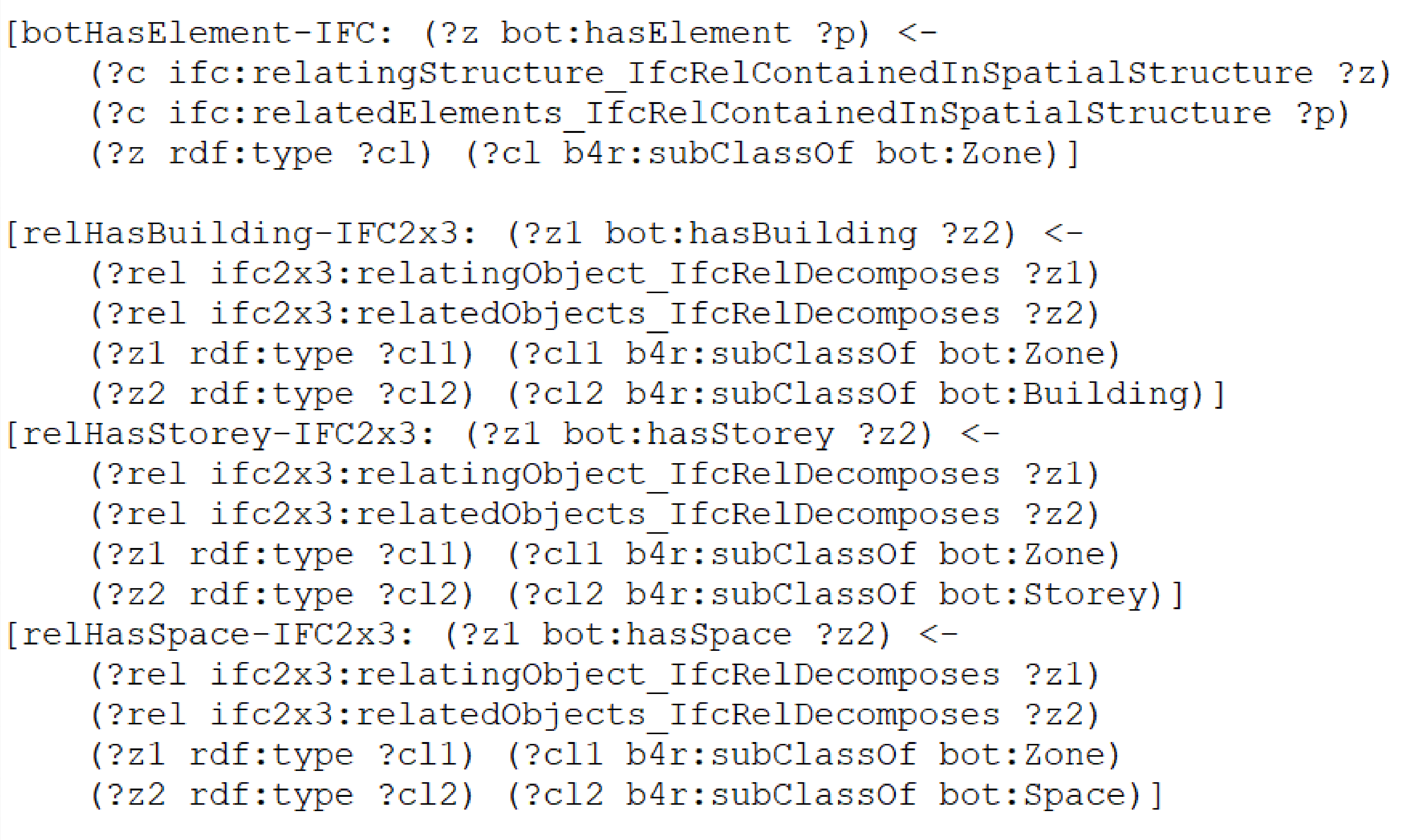


```
[equivalent1: (?a owl:equivalentClass ?b) -> (?a rdfs:subClassOf ?b) (?b rdfs:subClassOf ?a)]

-> table(b4r:subClassOf).
[transSubClassZone:  -> (bot:Zone b4r:subClassOf bot:Zone)]
[transSubClassSystemComponent:  -> (sot:SystemComponent b4r:subClassOf sot:SystemComponent)]
[transSubClassSystem:  -> (sot:System b4r:subClassOf sot:System)]
[transSubClass1: (?cls1 b4r:subClassOf ?cls2) <- (?cls1 rdfs:subClassOf ?cls2)]
[transSubClass2: (?cls1 b4r:subClassOf ?cls3) <- (?cls1 rdfs:subClassOf ?cls2) (?cls2 b4r:subClassOf ?cls3)]

-> table(list:isIn).
[isIn1: (?elt list:isIn ?l) <- (?l list:hasContents ?elt)]
[isIn2: (?elt list:isIn ?l) <- (?l list:hasNext ?queue) (?elt list:isIn ?queue)]
```

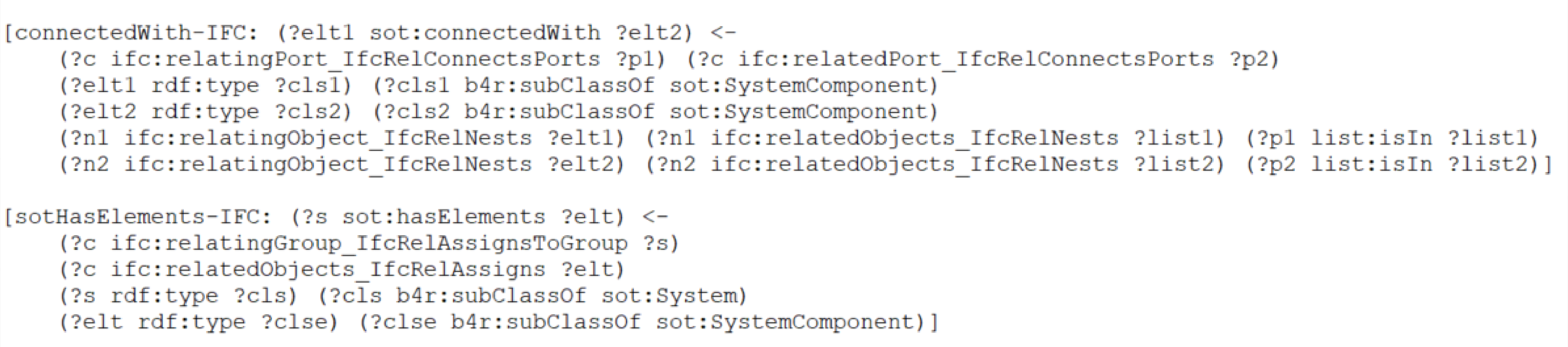
BOT rules



```
[botHasElement-IFC: (?z bot:hasElement ?p) <-
    (?c ifc:relatingStructure_IfcRelContainedInSpatialStructure ?z)
    (?c ifc:relatedElements_IfcRelContainedInSpatialStructure ?p)
    (?z rdf:type ?cl) (?cl b4r:subClassOf bot:Zone)]

[relHasBuilding-IFC2x3: (?z1 bot:hasBuilding ?z2) <-
    (?rel ifc2x3:relatingObject_IfcRelDecomposes ?z1)
    (?rel ifc2x3:relatedObjects_IfcRelDecomposes ?z2)
    (?z1 rdf:type ?cl1) (?cl1 b4r:subClassOf bot:Zone)
    (?z2 rdf:type ?cl2) (?cl2 b4r:subClassOf bot:Building)]
[relHasStorey-IFC2x3: (?z1 bot:hasStorey ?z2) <-
    (?rel ifc2x3:relatingObject_IfcRelDecomposes ?z1)
    (?rel ifc2x3:relatedObjects_IfcRelDecomposes ?z2)
    (?z1 rdf:type ?cl1) (?cl1 b4r:subClassOf bot:Zone)
    (?z2 rdf:type ?cl2) (?cl2 b4r:subClassOf bot:Storey)]
[relHasSpace-IFC2x3: (?z1 bot:hasSpace ?z2) <-
    (?rel ifc2x3:relatingObject_IfcRelDecomposes ?z1)
    (?rel ifc2x3:relatedObjects_IfcRelDecomposes ?z2)
    (?z1 rdf:type ?cl1) (?cl1 b4r:subClassOf bot:Zone)
    (?z2 rdf:type ?cl2) (?cl2 b4r:subClassOf bot:Space)]
```

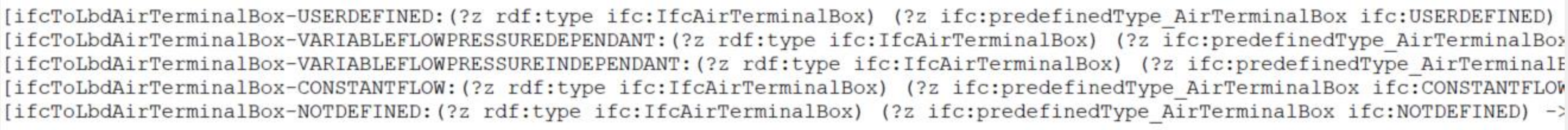
SOT rules



```
[connectedWith-IFC: (?elt1 sot:connectedWith ?elt2) <-
    (?c ifc:relatingPort_IfcRelConnectsPorts ?p1) (?c ifc:relatedPort_IfcRelConnectsPorts ?p2)
    (?elt1 rdf:type ?cls1) (?cls1 b4r:subClassOf sot:SystemComponent)
    (?elt2 rdf:type ?cls2) (?cls2 b4r:subClassOf sot:SystemComponent)
    (?n1 ifc:relatingObject_IfcRelNests ?elt1) (?n1 ifc:relatedObjects_IfcRelNests ?list1) (?p1 list:isIn ?list1)
    (?n2 ifc:relatingObject_IfcRelNests ?elt2) (?n2 ifc:relatedObjects_IfcRelNests ?list2) (?p2 list:isIn ?list2)]

[sotHasElements-IFC: (?s sot:hasElements ?elt) <-
    (?c ifc:relatingGroup_IfcRelAssignsToGroup ?s)
    (?c ifc:relatedObjects_IfcRelAssigns ?elt)
    (?s rdf:type ?cls) (?cls b4r:subClassOf sot:System)
    (?elt rdf:type ?clse) (?clse b4r:subClassOf sot:SystemComponent)]
```

Product layer rules



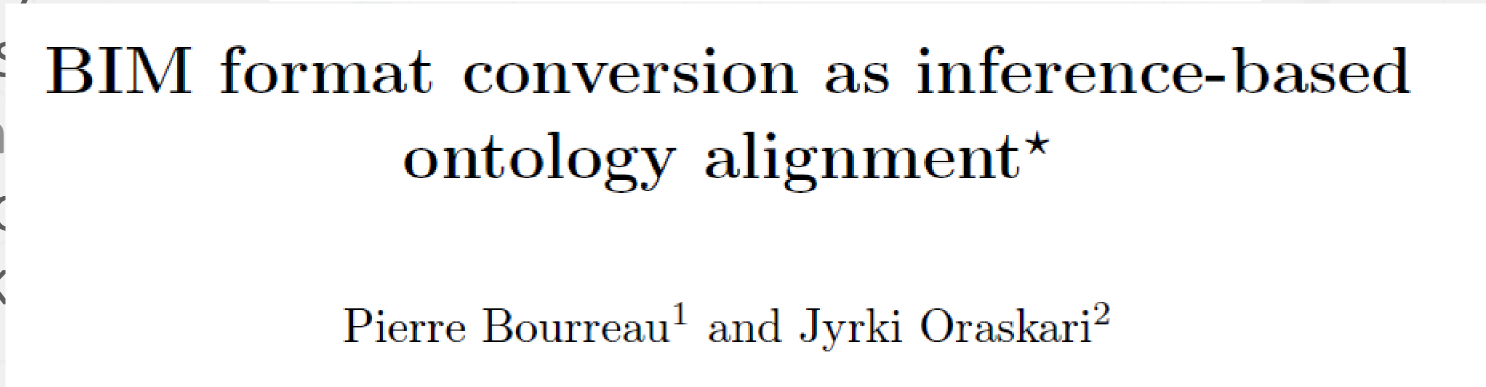
```
[ifcToLbdAirTerminalBox-USERDEFINED:(?z rdf:type ifc:IfcAirTerminalBox) (?z ifc:predefinedType_AirTerminalBox ifc:USERDEFINED)
[ifcToLbdAirTerminalBox-VARIABLEFLOWPRESSUREDEPENDANT:(?z rdf:type ifc:IfcAirTerminalBox) (?z ifc:predefinedType_AirTerminalBox
[ifcToLbdAirTerminalBox-VARIABLEFLOWPRESSUREINDEPENDANT:(?z rdf:type ifc:IfcAirTerminalBox) (?z ifc:predefinedType_AirTerminalB
[ifcToLbdAirTerminalBox-CONSTANTFLOW:(?z rdf:type ifc:IfcAirTerminalBox) (?z ifc:predefinedType_AirTerminalBox ifc:CONSTANTFLOW
[ifcToLbdAirTerminalBox-NOTDEFINED:(?z rdf:type ifc:IfcAirTerminalBox) (?z ifc:predefinedType_AirTerminalBox ifc:NOTDEFINED) ->
```

… + ~750 rules generated with a Python script

Properties

Some thoughts

- Multiplicity of on m
- Already a lot of t ntics (reasoning is one of them)
- Convers
  - Chan model
- Allows c nodel
- Remark t


BIM format conversion as inference-based ontology alignment*

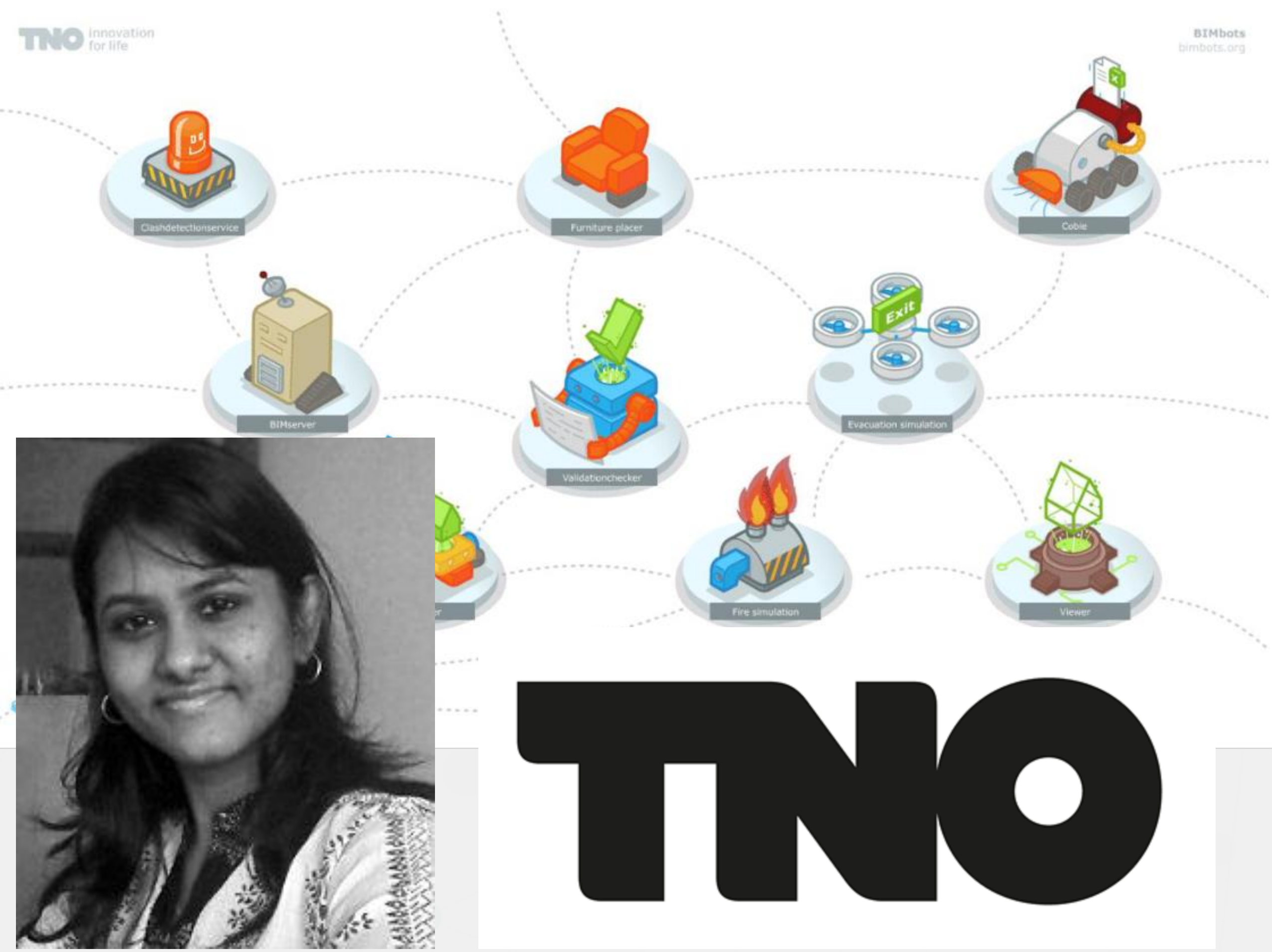
Pierre Bourreau[1] and Jyrki Oraskari[2]

- But no conversion on geometry
- BIM4Ren to ifcOwl: more complex (rules are not safe)
  - Use of `makeTemp()`

Some more things

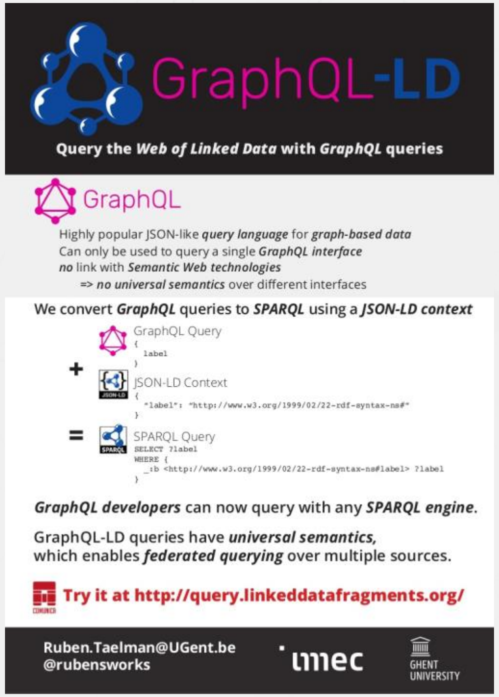
BIM Models checkers

- Checkers as (micro) web services
  - Micro service ≤ Web service
  - BIM Bots
- In BIM4Ren we use SHACL checkers
  - Semantic web micro services
  - Semantic BIM Bots
- Maestro Bot
  - Orchestration service



Data access

- SPARQL for free
  - Expert language
  - 0.X% of web service developers know it
- REST APIs
  - Easy to use
  - Kind of contract between server and client
  - Need to implement SPARQL Queries… tedious work (no ORM)
- GraphQL-LD
  - JS at either client of server
  - Restricted to queries (no edition) – still?
  - Kind of another expert language
- RAMOSE
  - Ease the deployment of REST APIs on top of SPARQL
  - Also restricted to Get
  - Need more work and contributions



GraphQL-LD

Query the *Web of Linked Data* with *GraphQL* queries

GraphQL

Highly popular JSON-like ***query language*** for ***graph-based data***
Can only be used to query a single ***GraphQL interface***
no link with ***Semantic Web technologies***
=> ***no universal semantics*** over different interfaces

We convert ***GraphQL*** queries to ***SPARQL*** using a ***JSON-LD context***

GraphQL Query
```
{
  label
}
```
+ JSON-LD Context
```
{
  "label": "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
}
```
= SPARQL Query
```
SELECT ?label
WHERE {
  _:b <http://www.w3.org/1999/02/22-rdf-syntax-ns#label> ?label
}
```

GraphQL developers can now query with any ***SPARQL engine***.

GraphQL-LD queries have ***universal semantics,***
which enables ***federated querying*** over multiple sources.

Try it at http://query.linkeddatafragments.org/

Ruben.Taelman@UGent.be
@rubensworks