

# Semantic Web (CO7516)

*(Assignment 2)*

## SPARQL queries and OWL ontologies

# Index

## Part I

<b>1- Examples of constraints and limited expressivity of the RDF Schema</b>	<b>2</b>
<b>2- Three SPARQL query for competency questions from your coursework 1.</b>	<b>2</b>
<b>i</b> Query 1 - - - - -	3
<b>ii</b> Query 2 - - - - -	4
<b>iii</b> Query 3 - - - - -	4
<b>3- Two SPARQL queries to search your ontology - - - - -</b>	<b>4</b>
<b>i</b> Query 1 - - - - -	4
<b>ii</b> Query 2 - - - - -	5

## Part 2

1- OWL subclass restrictions using owl:Restriction - - - - -	6
2. DataType Properties and at least 2 ObjectProperties. - - - - -	7
3. Property restrictions each on the domain and range. - - - - -	9
4. Class defined using owl:intersectionOf. - - - - -	10
5. Class defined using owl:unionOf. - - - - -	10
6. Cardinality restrictions (exact, min and max). - - - - -	11
7. Existential restrictions. - - - - -	14
<b>8. Universal restrictions. - - - - -</b>	<b>15</b>
<b>9. Properties</b>	
a. 2 Symmetric Properties - - - - -	16
b. 2 Transitive Properties - - - - -	17
c. 2 Inverse Properties - - - - -	18
d. 2 Functional Properties - - - - -	20
e. 2 Inverse Functional Properties - - - - -	23
f. 1 Reflexive Property - - - - -	24
g. 1 Irreflexive Property - - - - -	25
<b>10. Conclusion</b>	<b>25</b>
<b>11. References</b>	<b>25</b>

## PART 1

### 1- Give two examples of constraints that you were not able to describe in Coursework 1 due to the limited expressivity of the RDF Schema

#### 1. Example 1

In RDFS I could not restrict the instances of a “Materials” relates to a specific Construction Specifications Institute (CSI) “Division” using “belongstoDivision” and restrict it to exactly “1” Division.

$(\text{Material} \sqsubseteq \text{Division} \sqcap = 1 \text{ belongstoDivision})$

#### 2. Example 2

In RDFS I could not restrict Class, limited to certain domain and range, for example, contractor is a company who is “**BuilderOf**” some constructions either “residential” or “educational” buildings. In RDFs I could only define two subclasses “residential” and “educational” of the construction class.

$(\text{Contractor} \sqsubseteq \text{Company} \sqcap \exists \text{ builderof. Construction})$

### 2- Choose the three most important competency questions from your coursework 1. For each of those questions, write a SPARQL query to answers it.

**Query 1:** Who are the nearest suppliers of a specific building material near a construction site?

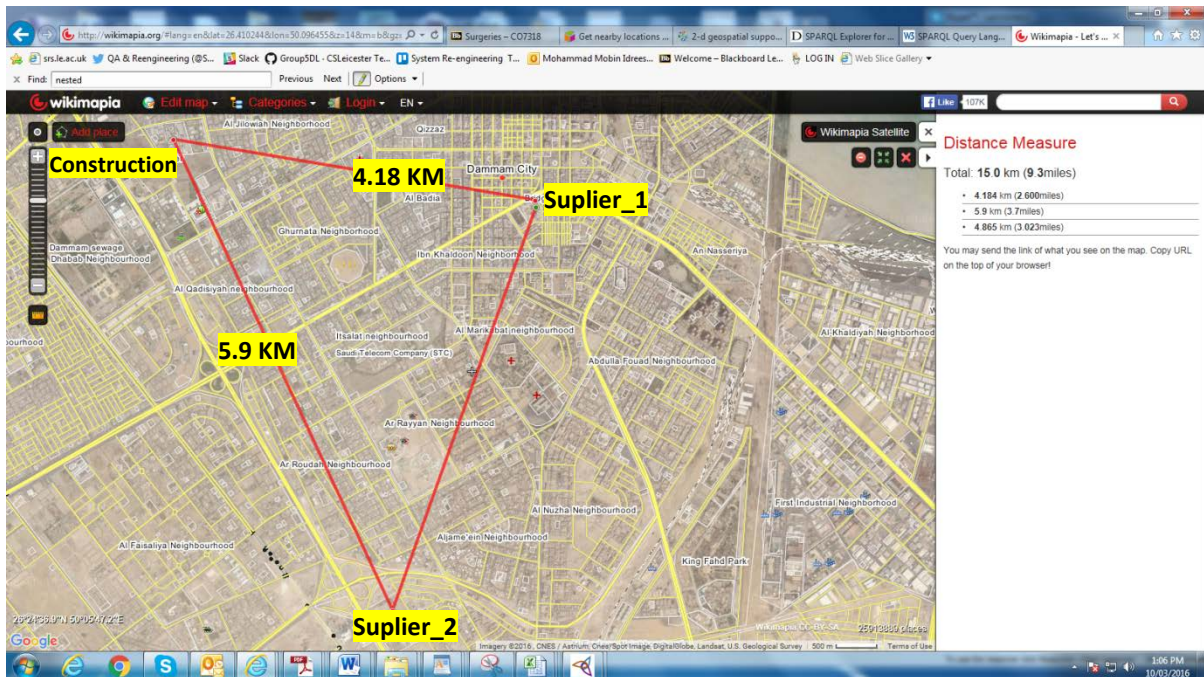
```
PREFIX scm: <http://www.Co7516-ontologies.com/GroupA/OntologyGroupA#>

SELECT DISTINCT ?Supplier ?Material ?Latitude ?Longitude (((?vertex_a-
?Latitude)*(?vertex_a-?Latitude) +(?vertex_b-?Longitude)*(?vertex_b-?Longitude)) AS
?distance)
  WHERE {
    ?Supplier scm:Latitude ?Latitude ;
              scm:Longitude ?Longitude.
    OPTIONAL { ?Supplier scm:Supplies?Material. FILTER
regex(str(?Material), "Nano_Concrete")}
    {
      SELECT ?vertex_a ?vertex_b
        WHERE {
          ?Latitude scm:Latitude ?vertex_a;
                    scm:Longitude ?vertex_b.
          FILTER (?Latitude = scm:University_of_Leicester)
        }
    }
  }
} LIMIT 50
```

## Results:

The “**Supplier\_1**” is the nearest supplier of “**Nano\_concrete**” from the construction site “University of Leicester”

Supplier	Material	Latitude	Longitude	distance
University_of_Leicester		"26.434687" ^ "50.059977" ^ "0.000000000000"		
Supplier_1	Nano_Concrete	"26.430152" ^ "50.100145" ^ "0.001634034449"		
Supplier_2	Nano_Concrete	"26.387333" ^ "50.090790" ^ "0.003191842285"		



**Query 2:** What material is best for a particular kind of construction?

```

SELECT DISTINCT ?Construction_Type ?Material ?toxic_level
WHERE {
    ?Construction_Type a scm:Residential ;
                      scm:usingMaterial ?Material .
    ?Material scm:hasToxicLevel ?toxic_level .
    FILTER (?toxic_level <=3) .
    OPTIONAL { ?Material scm:ManufacturedBy ?m .
    }
}
LIMIT 20

```

## Results

In this Example, the objective is to find building materials used in “**Residential**” type of construction that has Toxic Level below the provided level.

Construction_Type	Material	toxic_level
Residential_Building	Nano_Concrete	"3"^^<http://www.w3.org/2001/XMLSchema

**Query 3:** Which contractor has completed the most projects for a specific type of construction?

```
SELECT DISTINCT ?contractor ?category (COUNT(DISTINCT ?building) as
?Total_Projects)
WHERE{

    ?building scm:BuildBy ?contractor;
    a ?category.
    FILTER (?category = scm:Educational || ?category = scm:Residential)
}
GROUP BY ?contractor ?category

LIMIT 20
```

### Results

The contractor is grouped by “contractor” and “category”. The results show, “Consultant\_1” has completed “2” Residential” and “1” Educational” Project

contractor	category	Total_Projects
Contractor_2	Educational	"1"^^<http://www.w3.org/2001/XMLSchema
Contractor_1	Residential	"2"^^<http://www.w3.org/2001/XMLSchema

**3- Write two SPARQL queries to search your ontology, briefly explain what will be returned as a result. The queries should demonstrate ALL these features: OPTIONAL, FILTER, UNION, GROUP BY**

**Query 1:** Which construction companies have built houses for their own employees?

```
SELECT DISTINCT ?Employee ?Owner_of_Building ?Works_at_company
?Built_By_Contractor ?Location
WHERE {

    ?Employee scm:isOwnerOf ?Owner_of_Building ;
        scm:WorksAt ?Works_at_company .
    ?Owner_of_Building scm:BuildBy ?Built_By_Contractor .
    OPTIONAL { ?Built_By_Contractor scm:isLocated ?Location .}
    FILTER regex(str(?Employee), "anna","i") }
```

### Inference:

“Anna” owns a “Residential\_Building”, she works at “Contractor\_1”, the same company built her house. The inference is that the relation “BuildBy” between Building and contractor class. Here a FILTER is used to narrow the results to the keyword “anna”.

Employee	Owner_of_Building	Works_at_company	Built_By_Contractor
Anna_employee	Residential_Building	Contractor_1	Contractor_1

**Query 2:** Which buildings are using Materials of specific building Standards?

```
SELECT DISTINCT ?Division ?Material ?buildings ?SubDivision
WHERE {
  {
    ?Division a scm:Division ;
              scm:hasChemical ?Material .
    ?Material scm:usedIn ?buildings .
  }
  UNION
  {
    ?Division a scm:Division ;
              scm:hasMechanical ?Material .
    ?Material scm:usedIn ?buildings .
  }
}
ORDER BY ?Division
```

### Inference:

The “UNION” query merges the “CSI-Div1” and “CSI-Div2” for the “Chemical” and “Mechanical” building materials. Here there “Mobin Building” is using “Chemical\_Material\_1” that belongs to “Div-1” of The CSI standard, and “University of Leicester” is using “Mechanical\_Material\_1” which belongs to Div-2.

Division	Material	buildings	SubDivision
Div1	Chemical_Material_1	Mobin_Building	
Div1	Nano_Concrete	Residential_Building	
Div2	Mechanical_Material_1	University_of_Leicester	

## PART 2

### 1- Include at least 4 OWL subclass restrictions (use owl:Restriction).

- 1- The concept Contractor is *Subclass-Of* “Company” restricted to the range “**BuilderOf**” of between Contractor and Construction.

```
:Contractor rdf:type owl:Class ;  
    rdfs:subClassOf :Company ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :BuilderOf ;  
          owl:allValuesFrom :Construction] .
```

Instance:

Contractor\_1 **builderOf** University of Leicester

```
:Contractor_1 rdf:type :Company ,  
    owl:NamedIndividual ;  
    :BuilderOf :Residential_Building ,  
    :University_of_Leicester .
```

- 2- The Manufacturer Subclass-Of “Company” can be a “ManufacturerOf” of only Material.

```
:Manufacturer rdf:type owl:Class ;  
    rdfs:subClassOf :Company ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :ManufacturerOf ;  
          owl:allValuesFrom :Material] .
```

Instance:

Manufacturer\_1 **ManufacturerOf** Electrical\_Material\_1 and Nano\_Concrete.

```
:Manufacturer_1 rdf:type :Company ,  
    owl:NamedIndividual ;  
    :ManufacturerOf :Electrical_Material_1 ,  
    :Nano_Concrete .
```

- 3- The Supplier *Subclass-Of* “Company” can be “**SupplierOf**” of only Materials.

```
:Supplier rdf:type owl:Class ;  
    rdfs:subClassOf :Company ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :SupplierOf ;  
          owl:allValuesFrom :Material] .
```

Instance:

Supplier\_1 **SupplierOf** Nano\_Concrete

```
:Supplier_1 rdf:type :Company ,  
    owl:NamedIndividual ;  
    :SupplierOf :Nano_Concrete .
```

- 4- Div-1 *Subclass-Of* concept “**Division**” can have only “**Chemical Material**”, Div-2 range is restricted to “hasMechanical” materials and “Div-3” has range “Chemical” Materials.

```
:Div1 rdf:type owl:Class ;  
      rdfs:subClassOf :Division ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :hasChemical ;  
          owl:allValuesFrom :Chemical ] .
```

Instance:

Div\_1 **hasChemical** Chemical\_Material\_1 and Nano\_Concrete.

```
:Div1 rdf:type :Division ,  
      owl:NamedIndividual ;  
      :hasChemical :Chemical_Material_1 ;  
      :hasChemical :Nano_Concrete .
```

## 2- Include at least 2 DataType Properties and at least 2 ObjectProperties.

### Data property -1

The purpose of object property is to link resources to literals.

“**Product\_code**” is unique key for Materials, its using literal “string”.

```
:Product_Code rdf:type owl:DatatypeProperty ;  
      rdfs:domain :Material ;  
      rdfs:range xsd:string .
```

Instance

“Nano\_Concrete” has **Product\_Code** “**NC-001**”.

```
:Nano_Concrete rdf:type :Material ,  
      owl:NamedIndividual ;  
      :Product_Code “NC-001”^^xsd:string ;
```

### Data property -2

“**Title\_No**” is unique key for Company or Construction, literal type used is “string”

```
:Title_No rdf:type owl:DatatypeProperty ;  
      rdfs:domain :Company ,  
        :Construction ;  
      rdfs:range xsd:string .
```

“Residential\_Building” has title No. “**TN:456PWZ**”.

```
:Residential_Building rdf:type :Residential ,  
      owl:NamedIndividual ;  
      :Title_No “TN:456PWZ”^^xsd:string ;
```



### Object Property -1 : hasMaterial

The purpose of object property is to link object to other objects . Construction materials belongs to an specific CSI Division. The Object Property **hasChemical** is a sub property of **hasMaterial**. Further sub-properties are “**hasElectrical**” and **hasMechanical**”.

```
:hasChemical rdf:type owl:ObjectProperty ;  
    dfs:domain :Division ;  
    rdfs:range :Chemical ;  
    rdfs:subPropertyOf :hasMaterial .
```

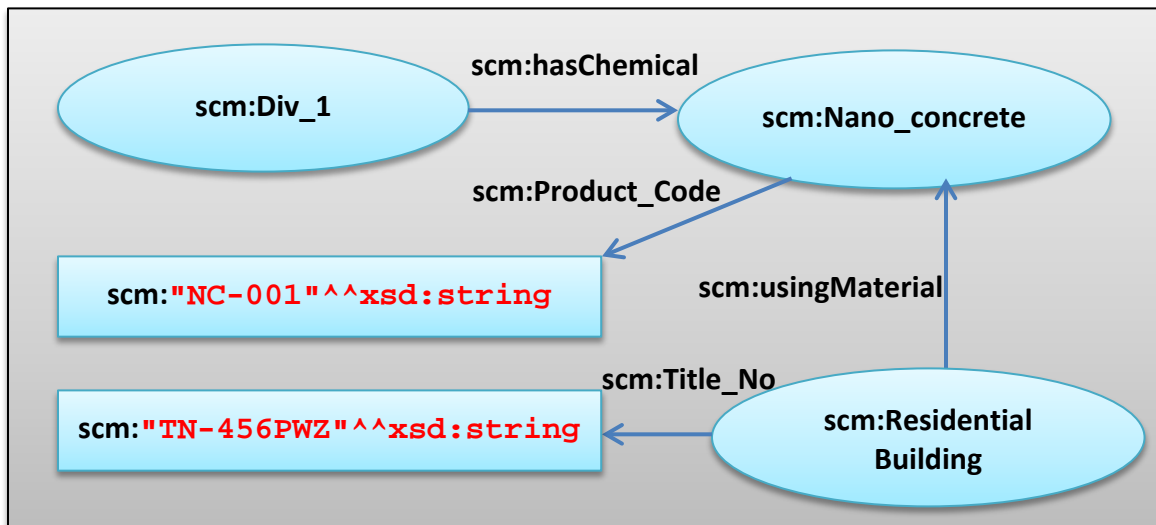
### Object Property 2: UsingMaterial

Buildings use specific sets of materials.

```
:Residential_Building rdf:type :Residential ,  
    owl:NamedIndividual ;  
    :usingMaterial :Nano_Concrete .
```

### Instance

The inference is that an instance of “residential building” is using “The Nano-Concrete” material which belongs to Div\_1 and it’s a Chemical.



3- Include at least 2 property restrictions each on the domain and range.

- 1- The “Person” can “StudyIn” in Construction Type “Educational” and “LivesIn” Construction Type “Residential”.

```
:Educational rdf:type owl:Class ;
    rdfs:subClassOf :Construction ,
        [ rdf:type owl:Restriction ;
          owl:onProperty :StudyIn ;
          owl:allValuesFrom :Person ] ;
:Residential rdf:type owl:Class ;
    rdfs:subClassOf :Construction ,
        [ rdf:type owl:Restriction ;
          owl:onProperty :livesIn ;
          owl:someValuesFrom :Person] .
```

Instances:

Anna\_employee livesIn Residential\_Building BuildBy Contractor\_1 and usingMaterial Nano\_Concrete material.

```
:Residential_Building rdf:type :Residential ,
    owl:NamedIndividual ;
    :livesIn :Anna_employee ;
    :BuildBy :Contractor_1 ;
    :usingMaterial :Nano_Concrete .
```

Peter studyIn University\_of\_Leicester.

```
:University_of_Leicester rdf:type :Educational ,
    owl:NamedIndividual ;
    :StudyIn :Peter .
```

- 2- The concept Manager is *Subclass-Of* “Person” restricted to the range “ManagerOf” of Company.

```
:Manager rdf:type owl:Class ;
    rdfs:subClassOf :Person ,
        [ rdf:type owl:Restriction ;
          owl:onProperty :ManagerOf ;
          owl:someValuesFrom :Company] ;
```

Instances

John\_Manager is ManagerOf Contractor\_2 and he isOwnerOf Mobin\_Building.

```
:John_Manager rdf:type :Person ,
    owl:NamedIndividual ;
    :ManagerOf :Contractor_2 ;
    :isOwnerOf :Mobin_Building .
```

#### 4- Include at least 1 Class defined using owl:intersectionOf.

Electro-Mechanical materials are intersection of Electrical AND Mechanical Material

```
:Electro_Machanical rdf:type owl:Class ;  
  owl:equivalentClass [ rdf:type owl:Class ;  
    owl:intersectionOf ( :Electrical :Mechanical ) ] ;  
  rdfs:subClassOf :Material .
```

A Manager is a Person AND ManagerOf some Company.

```
:Manager rdf:type owl:Class ;  
  owl:equivalentClass [ rdf:type owl:Class ;  
    owl:intersectionOf ( :Person  
      [ rdf:type owl:Restriction ;  
        owl:onProperty :ManagerOf ;  
          owl:someValuesFrom :Company ] ) ] ;
```

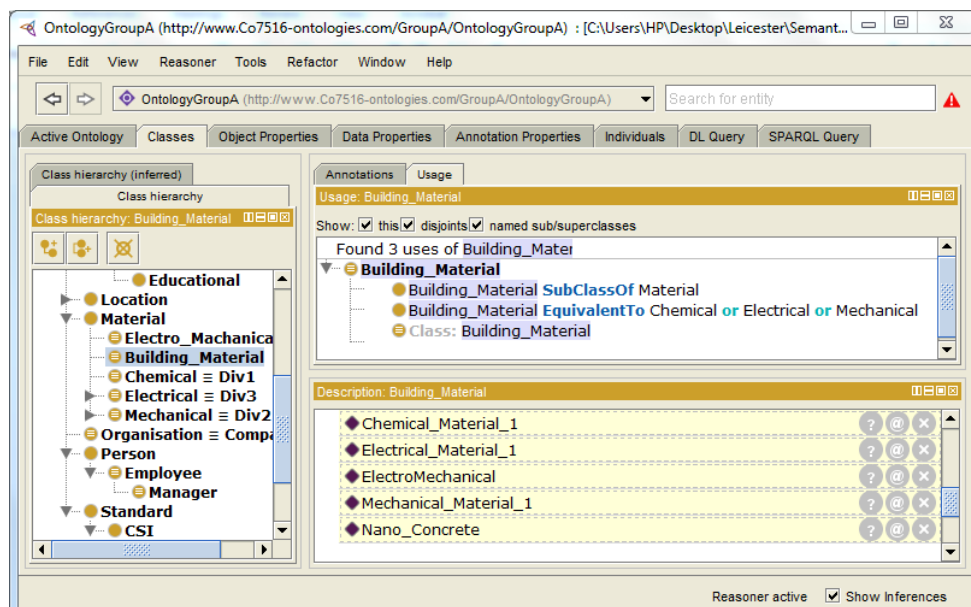
#### Instance

```
:John_Manager rdf:type :Person ,  
  owl:NamedIndividual ;  
  :ManagerOf :Contractor_2 ;
```

#### 5- Include at least 1 Class defined using owl:unionOf.

The concept “Building Materials” is union of “chemical”, electrical, and mechanical materials.

```
:Building_Material rdf:type owl:Class ;  
  owl:equivalentClass [ rdf:type owl:Class ;  
    owl:unionOf ( :Chemical:Electrical :Mechanical ) ] ;  
  rdfs:subClassOf :Material .
```



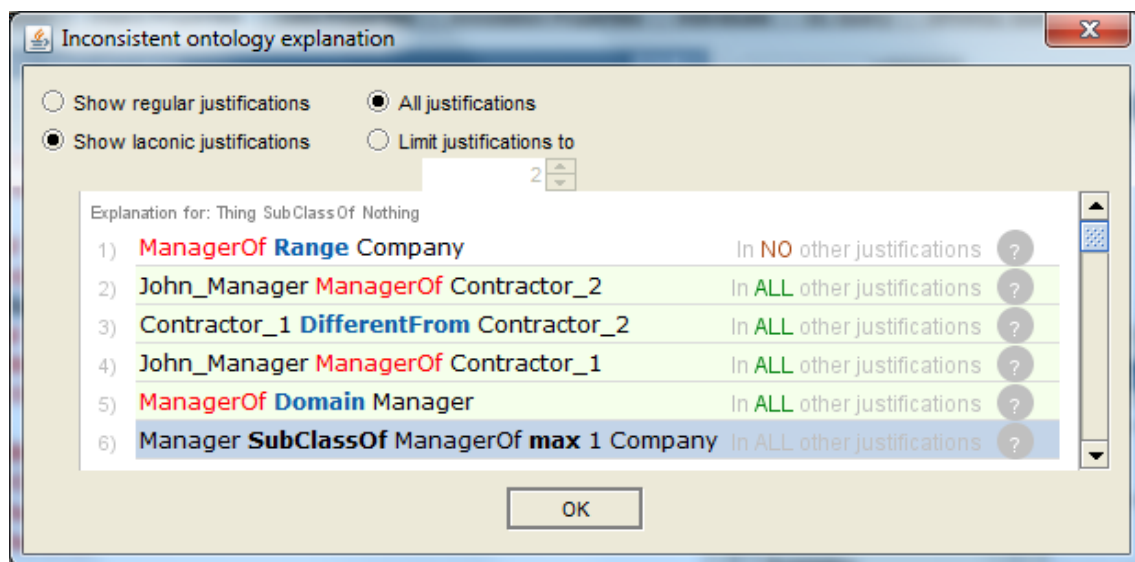
6- Include at least 3 Cardinality restrictions (exact, min and max).

(1) A Manager can be a Manager of Exactly one company.

```
:ManagerOf rdf:type owl:ObjectProperty ;  
    rdfs:range :Company ;  
    rdfs:domain :Manager ;  
    rdfs:range [ rdf:type owl:Restriction ;  
        owl:onProperty :ManagerOf ;  
        owl:onClass :Company ;  
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ] .
```

Instance

```
:John_Manager rdf:type :Person ,  
    owl:NamedIndividual ;  
    :ManagerOf :Contractor_1 ,  
    :Contractor_2 ;
```



# General axioms

The protégé ontology represents real world ontology, therefore **contractor\_1** are same as “**contractor\_2**” unless if we define them as distinct members

```
[ rdf:type owl:AllDifferent ;  
    owl:distinctMembers ( :Contractor_1  
        :Contractor_2  
    )  
] .
```

(2) A construction material can belongs to a single CSI Division.

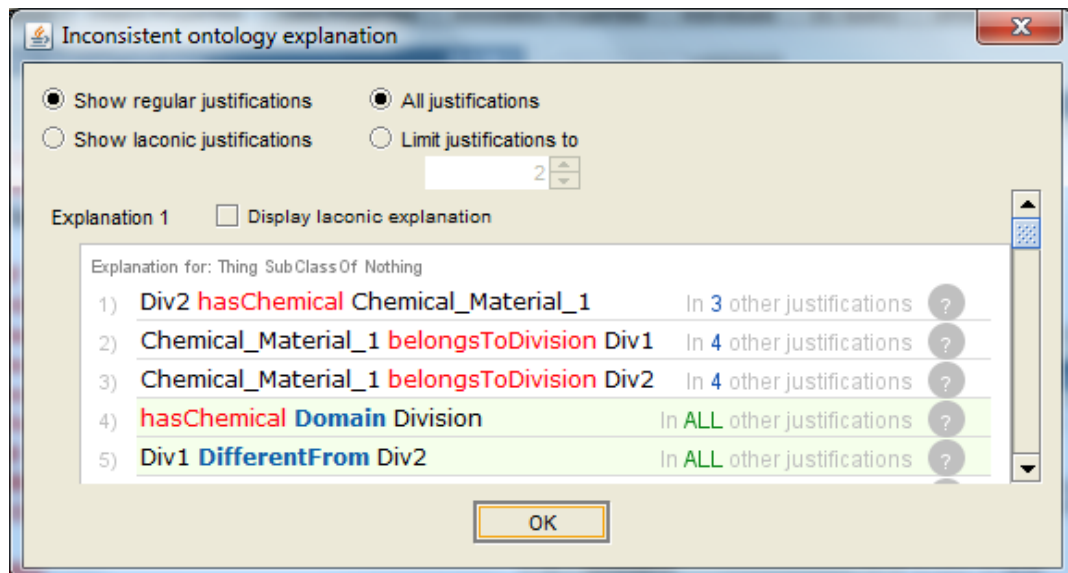
```
:belongsToDivision rdfs:type owl:ObjectProperty ;  
    rdfs:domain :Material ;  
    owl:inverseOf :hasMaterial ;  
    rdfs:range [ rdfs:type owl:Restriction ;  
        owl:onProperty :belongsToDivision ;  
        owl:onClass :Division ;  
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger  
    ] .
```

Instance

```
:Chemical_Material_1 rdfs:type :Material ,  
    owl:NamedIndividual ;  
  
    :belongsToDivision :Div1 ,  
        :Div2 ;
```

# General axioms

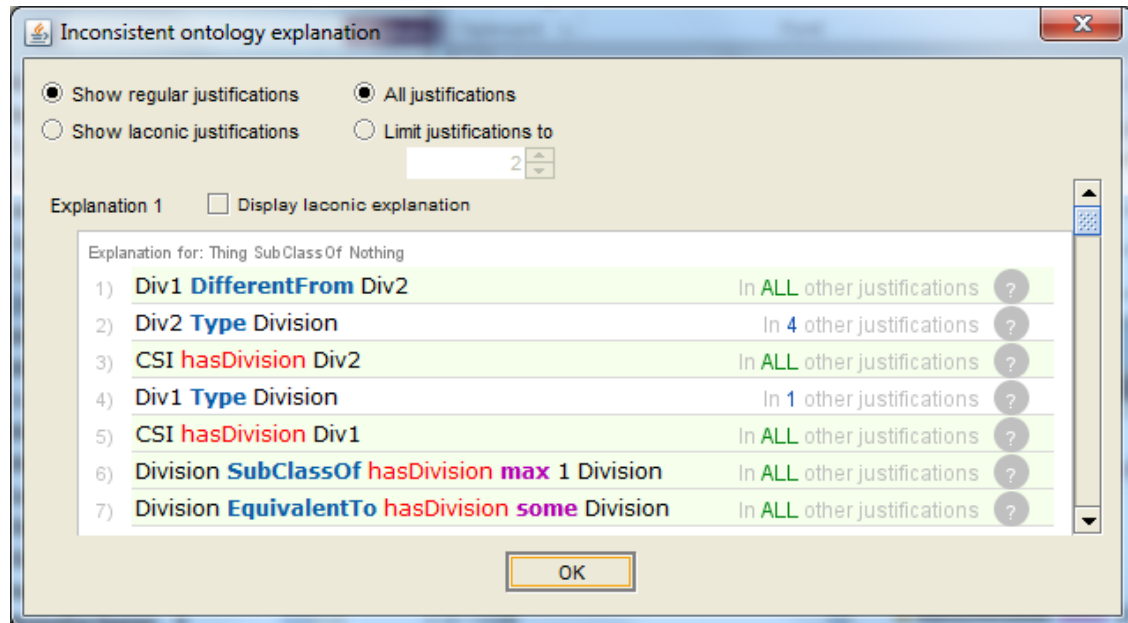
```
[ rdfs:type owl:AllDifferent ;  
    owl:distinctMembers ( :Div1  
        :Div2  
    )  
] .
```



### (3) Min and Max Cardinality Restriction

The MasterFormat of the Construction Specifications Institute (CSI) and Construction Specifications Canada (CSC) contains maximum 50 Divisions. [Reference](#).

```
:Division rdf:type owl:Class ;  
  
    rdfs:subClassOf :CSI ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :hasDivision ;  
          owl:onClass :Division ;  
          owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger  
        ] ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :hasDivision ;  
          owl:onClass :Division ;  
          owl:maxQualifiedCardinality "16"^^xsd:nonNegativeInteger  
        ] ,  
        [ rdf:type owl:Restriction ;  
          owl:onProperty :hasDivision ;  
          owl:allValuesFrom :Division  
        ] .
```



7- Include at least 2 Existential restrictions.

1- (DesignerOf **some** (Warehouse or Building or Educational))

The class “**DesignerOf**” consists of the individuals that are involved in the “**DesignOf**” “warehouses, residential or educational buildings..

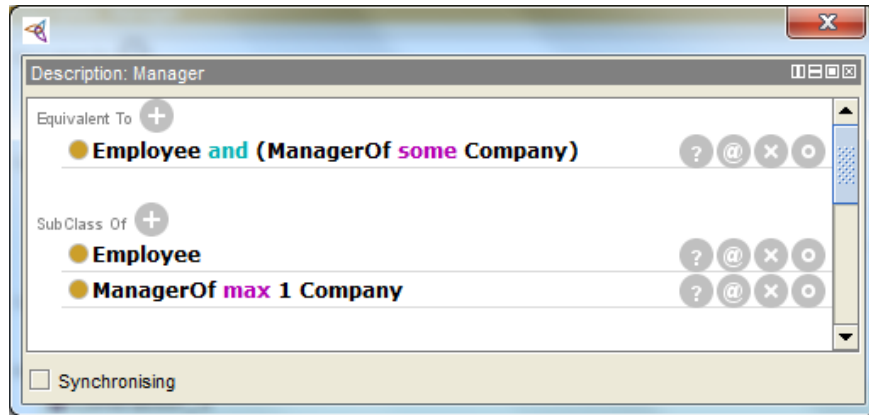
```
:Designer rdf:type owl:Class ;  
    owl:equivalentClass [ rdf:type owl:Restriction ;  
        owl:onProperty :DesignerOf ;  
        owl:someValuesFrom [ rdf:type owl:Class ;  
            owl:unionOf ( :Educational :Residential :Warehouse )
```



2- (Employee and (Manager of **some** Company))

A Manager Class consists of individuals who are linked to company instances using “**ManagerOf**” relationship and are type of **employee** class.

```
:Manager rdf:type owl:Class ;  
    owl:equivalentClass [ rdf:type owl:Class ;  
        owl:intersectionOf ( :Employee  
            [ rdf:type owl:Restriction ;  
                owl:onProperty :ManagerOf ;  
                owl:someValuesFrom :Company ] ) ] ;
```

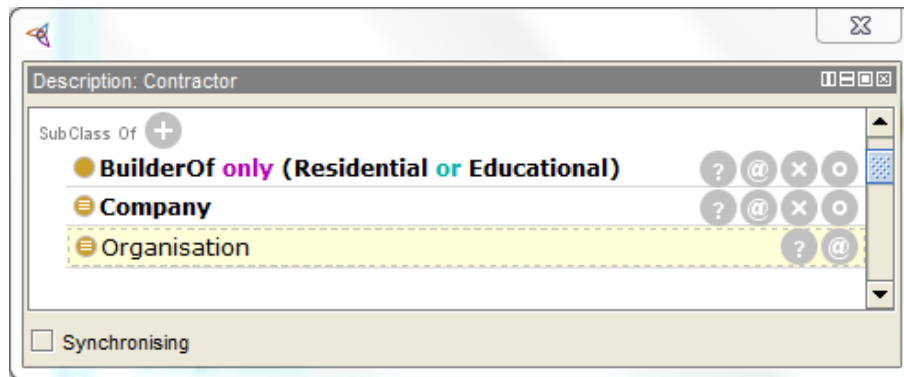


8- Include at least 2 Universal restrictions.

1- BuilderOf **only** (Residential and Educational)

A contractor is a builder of only constructions, this may includes only construction type, Residential or Educational.

```
:Contractor rdf:type owl:Class ;
    rdfs:subClassOf :Company ,
    [ rdf:type owl:Restriction ;
      owl:onProperty :BuilderOf ;
      owl:allValuesFrom [ rdf:type owl:Class ;
        owl:unionOf ( :Educational:Residential) ] ] .
```



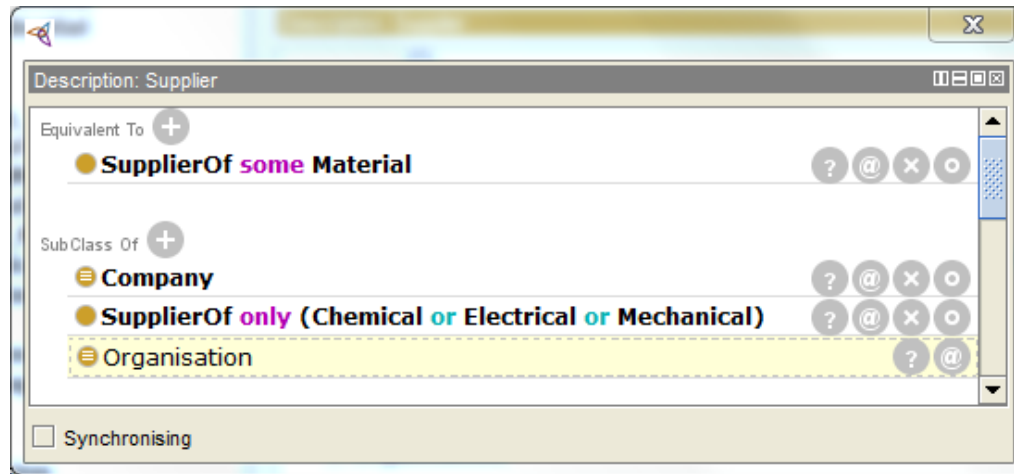
2- SupplierOf **only** (Chemical or Electrical or Mechanical)

A supplier is a supplier of only construction materials, this may includes only material type, Electrical, Mechanical or Chemical.

```
:Supplier rdf:type owl:Class ;
    rdfs:subClassOf :Company ,
    [ rdf:type owl:Restriction ;
      owl:onProperty :SupplierOf ;
      owl:allValuesFrom [ rdf:type owl:Class ;
```



```
owl:unionOf ( :Chemical:Electrical :Mechanical )]] .
```



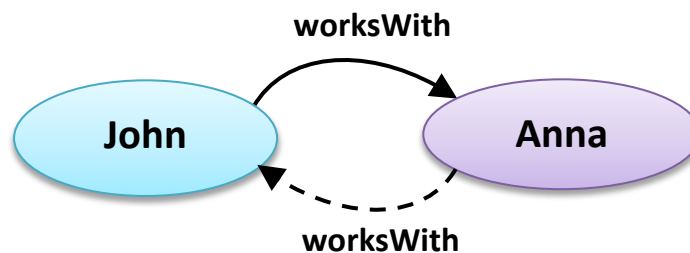
9- Include 2 symmetric, 2 transitive, 2 inverse, , 2 Functional, 2 inverse functional, 1 reflexive, 1 Irreflexive

#### (a) Symmetric Property

##### Example 1 : worksWith

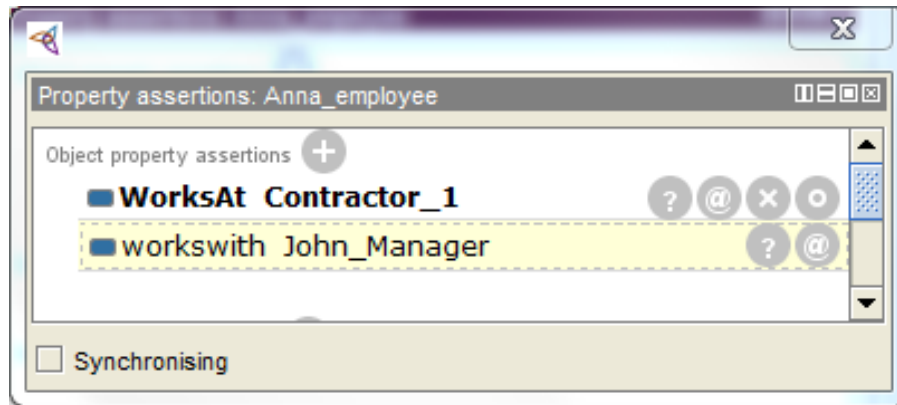
The relation “**p**” is symmetric If “**s**” relates to “**o**” then “**o**” relates to “**s**”. For instance, “workswith” relation is “**symmetric**” on “**Person**”. If Person “**A**” workswith Person “**B**”, then Person “**B**” workswith Person “**A**”.

```
<owl:ObjectProperty rdf:about="&scm:#worksWith">
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
  <rdfs:range rdf:resource="&scm:#Employee"/>
  <rdfs:domain rdf:resource="&scm:#Employee"/>
</owl:ObjectProperty>
```



##### Inference:

The Relation “worksWith” has been defined as Symmetric, we can infer that: Anna workswith John.

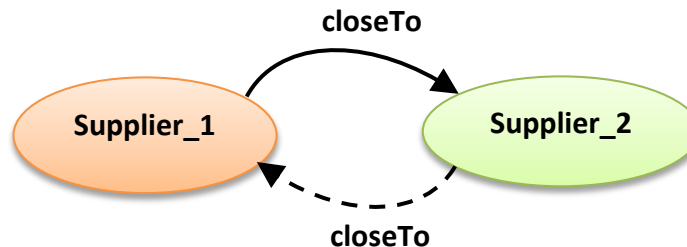


“Anna” worksWith “John”.

### Example 2: CloseTo

The relationship “CloseTo” on domain “Supplier” and range “Supplier” is symmetric.

```
<owl:ObjectProperty rdf:about="&scm:#closeTo">
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
  <rdfs:domain rdf:resource="&scm:#Supplier"/>
  <rdfs:range rdf:resource="&scm:#Supplier"/>
</owl:ObjectProperty>
```



### Inference

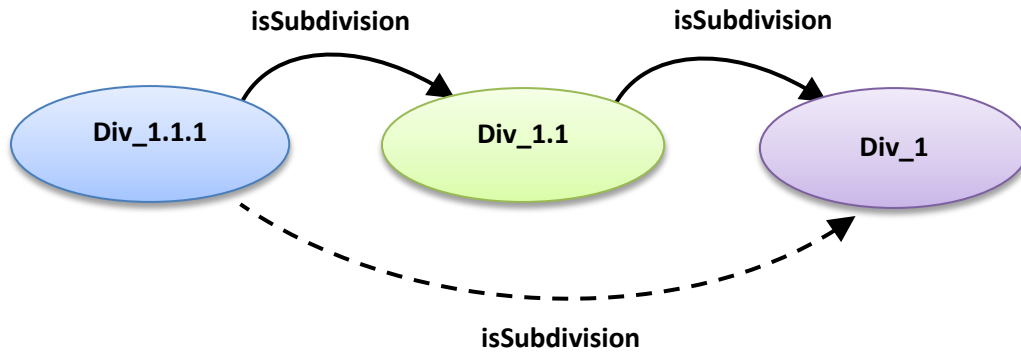
If **Supplier\_1** is **closeTo** **Supplier\_2**, then **Supplier\_2** is also **closeTo** **Supplier\_1**

### (b) Transitive Properties

#### Example 1: isSubDivision , hasSubDivision

The relationship “isSubDivision” on domain “division” and range “subDivision” is transitive.

```
<owl:ObjectProperty rdf:about="&scm:#isSubDivision">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:range rdf:resource="&scm:#Division"/>
  <rdfs:domain rdf:resource="&scm:#SubDivision"/>
</owl:ObjectProperty>
```



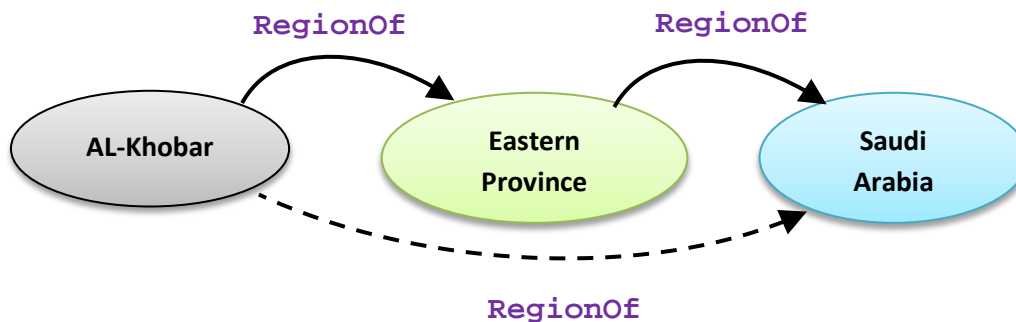
### Inference

If Div\_1.1 **isSubDivision** of Div\_1 and Div\_1.1.1 **isSubDivision** of Div\_1.1 then Div\_1.1.1 is also **isSubDivision** of Div\_1.

### Example 1: RegionOf

The relationship 'RegionOf' on domain "Location" and range "Location" is transitive.

```
<owl:ObjectProperty rdf:about="&scm;#RegionOf">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:domain rdf:resource="&scm;#Location"/>
  <rdfs:range rdf:resource="&scm;#Location"/>
</owl:ObjectProperty>
```



### Inference

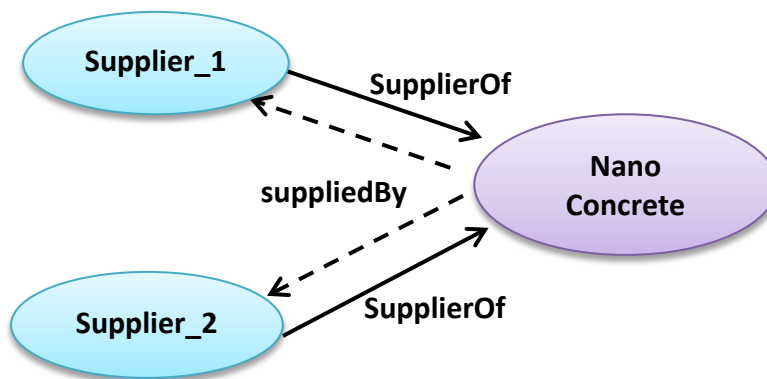
If 'Al-Khobar' is a RegionOf of 'Eastern Province' and 'Eastern Province' is a RegionOf 'Saudi Arabia' then 'Al-Khobar' is also a RegionOf of 'Saudi Arabia'.

### (c) Inverse Property

#### Example 1: SupplierOf , SuppliedBy

The relationship ‘SupplierOf’ and ‘SuppliedBy’ on domain “Supplier” and range “Material” is inverse.

```
<owl:ObjectProperty rdf:about="&scm:#SupplierOf">
  <rdfs:range rdf:resource="&scm:#Material"/>
  <owl:inverseOf rdf:resource="&scm:#SuppliedBy"/>
  <rdfs:domain rdf:resource="&scm:#Supplier"/>
</owl:ObjectProperty>
```



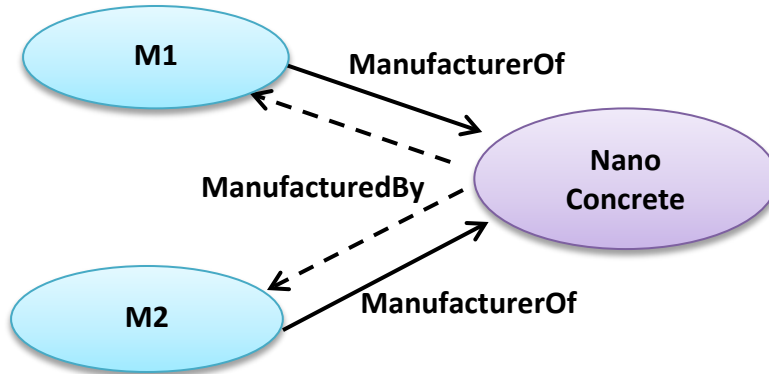
#### Inference

If Supplier\_1 is supplierOf Nano\_Concrete, then Nano\_material is suppliedBy Supplier\_1

#### Example 2: ManufacturerOf, ManufacturedBy

The relationship ‘ManufacturerOf’ and ‘ManufacturerBy’ on domain “manufacturer” and range “Material” is inverse.

```
<owl:ObjectProperty rdf:about="&scm:#ManufacturerOf">
  <rdfs:range rdf:resource="&scm:#Material"/>
  <owl:inverseOf rdf:resource="&scm:#ManufacturedBy"/>
  <rdfs:domain rdf:resource="&scm:#Manufacturer"/>
</owl:ObjectProperty>
```



### Inference

If **M1** is a manufacturer, **ManufacturerOf** Nano\_Concrete, then Nano\_material is **ManufacturedBy** M1

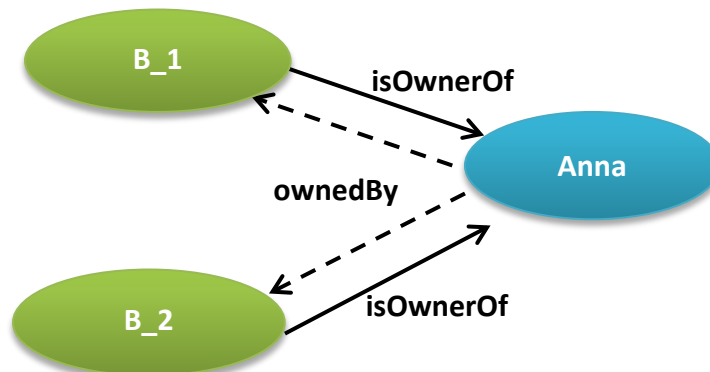
### (c) Functional Property

#### Example 1: **isOwnerOf** , **OwnedBy**

The relationship 'isOwnerOf' on domain "Person" and range "Residential" could be Functional. Which means that a Person can have at most one House (a good example of human economic dependence of ontologies):

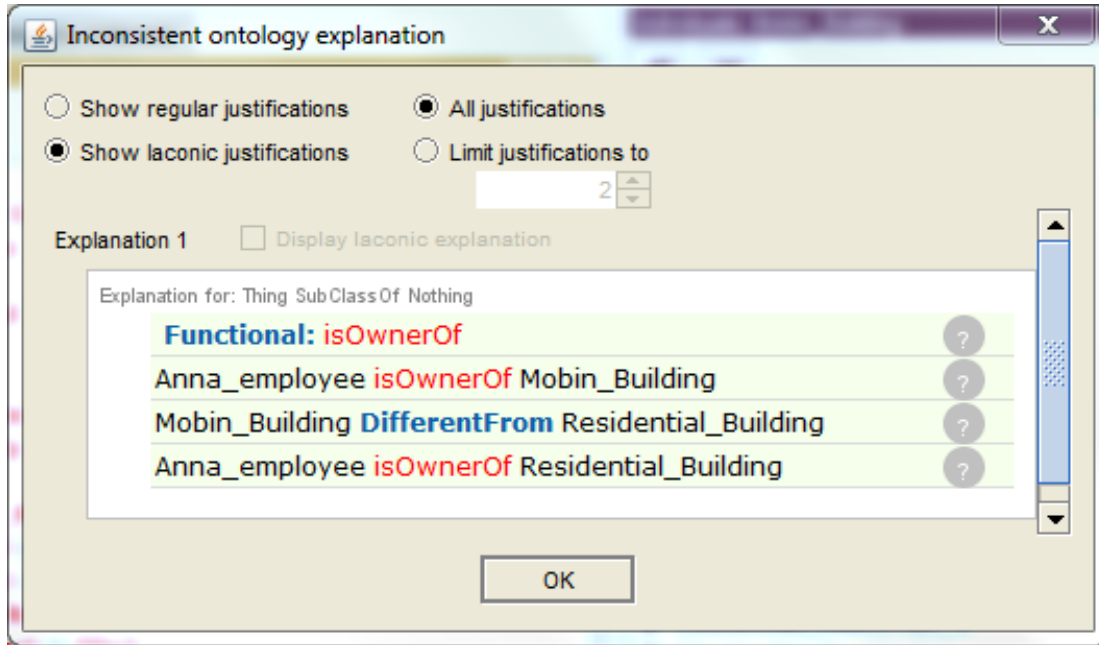
```

<owl:ObjectProperty rdf:about="&scm:#isOwnerOf">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:domain rdf:resource="&scm:#Person"/>
  <rdfs:range rdf:resource="&scm:#Residential"/>
  <owl:inverseOf rdf:resource="&scm:#isOwnedBy"/>
</owl:ObjectProperty>
  
```



## Inference

If **Anna** is the **OwnerOf 'B\_1'**, then Anna cannot be an ownerOf **B\_2** or any other instance of Residence.



### # General axioms

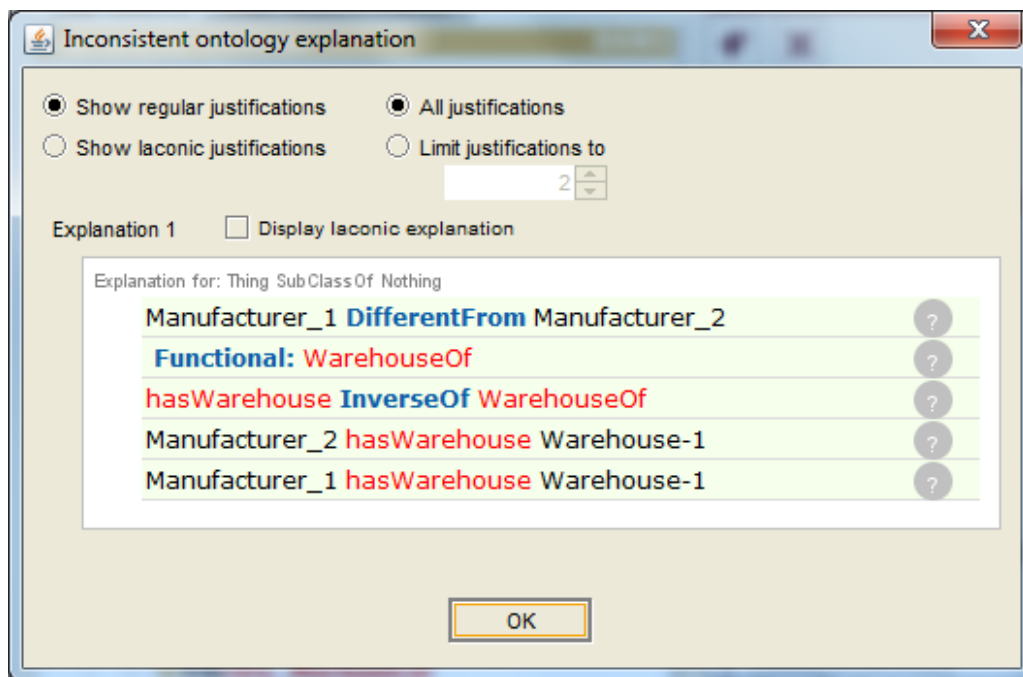
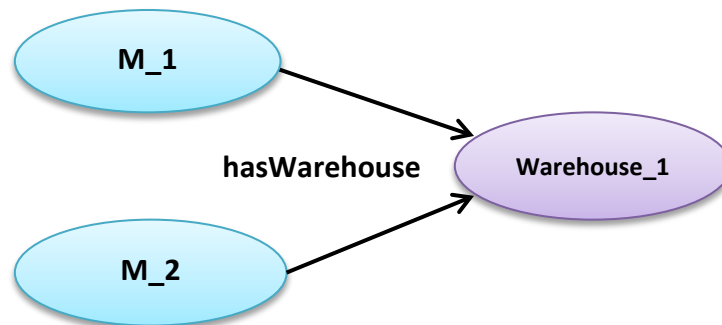
The protégé ontology represents real world ontology, therefore 'Mobin\_Building' is same as 'Residential\_Building' unless we define them as distinct members

```
[ rdf:type owl:AllDifferent ;  
  owl:distinctMembers ( : Mobin_Building  
                        : Residential_Building  
                        )  
].
```

### Example 2: WarehouseOf , hasWarehouse

Assumption is that a warehouse is unique to the companies, the relationship 'warehouseOf' on domain "Warehouse" and range "Company" could be Functional. Which means that a 'warehouse' can belong to only one company (either supplier or manufacturer).

```
<owl:ObjectProperty rdf:about="&scm:# WarehouseOf">  
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>  
  <rdfs:domain rdf:resource="&scm:#Warehouse"/>  
  <rdfs:range rdf:resource="&scm:#Company"/>  
  <owl:inverseOf rdf:resource="&scm:#hasWarehouse"/>
```



### Inference

If **Manufacturer\_1** has a **Warehouse\_1** then this warehouse can not belongs to **Manufacturer\_2** or any other instance of companies.

#### # General axioms

In real world ontology, '**Manufacturer\_1**' and '**Manufacturer\_2**' are same unless we define them as distinct as follows

```

[ rdf:type owl:AllDifferent ;
  owl:distinctMembers ( : Manufacturer_1
                        : Manufacturer_2
                        )
].

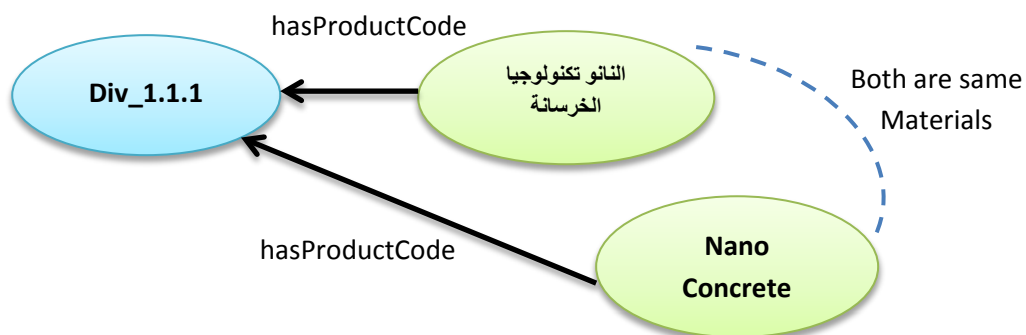
```

#### (d) Inverse Functional Property

##### Example 1: [hasProductCode](#)

If the relation is inverse-functional, then the objects uniquely determines the subject, for instance if a material has a product code then all the materials having that product code will be the same individuals.

```
<owl:ObjectProperty rdf:about="&scm:#hasProductCode">
  <rdf:type rdf:resource="&owl:InverseFunctionalProperty"/>
  <rdfs:range rdf:resource="&scm:#Division"/>
  <rdfs:domain rdf:resource="&scm:#Material"/>
</owl:ObjectProperty>
```



#### Inference

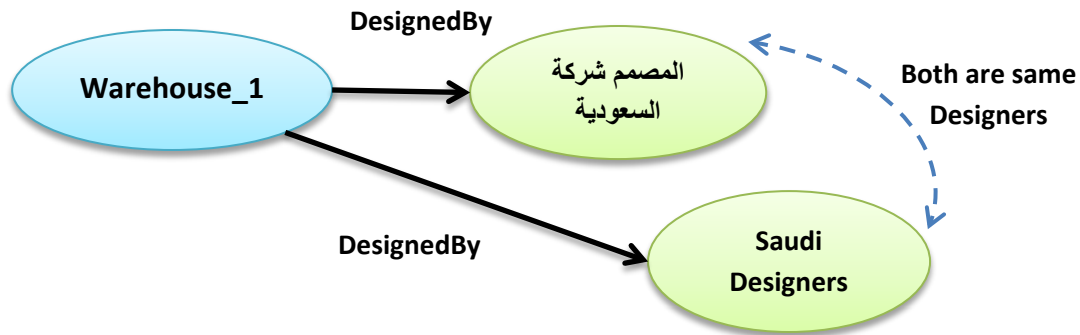
If 'Nano\_concrete' has product code 'Div\_1.1.1' and 'الخرسانة النانو تكنولوجيا' (Nano concrete in Arabic Language) also has product code Div\_1.1.1 then 'الخرسانة النانو تكنولوجيا' and 'Nano\_concrete' are same individuals.

##### Example 1: [DesignerOf](#)

Assume that a construction is designedBy only one designer company, if a construction is designedBy a company and the same construction is designedBy a designer company having different name, then both designer companies are same individuals.

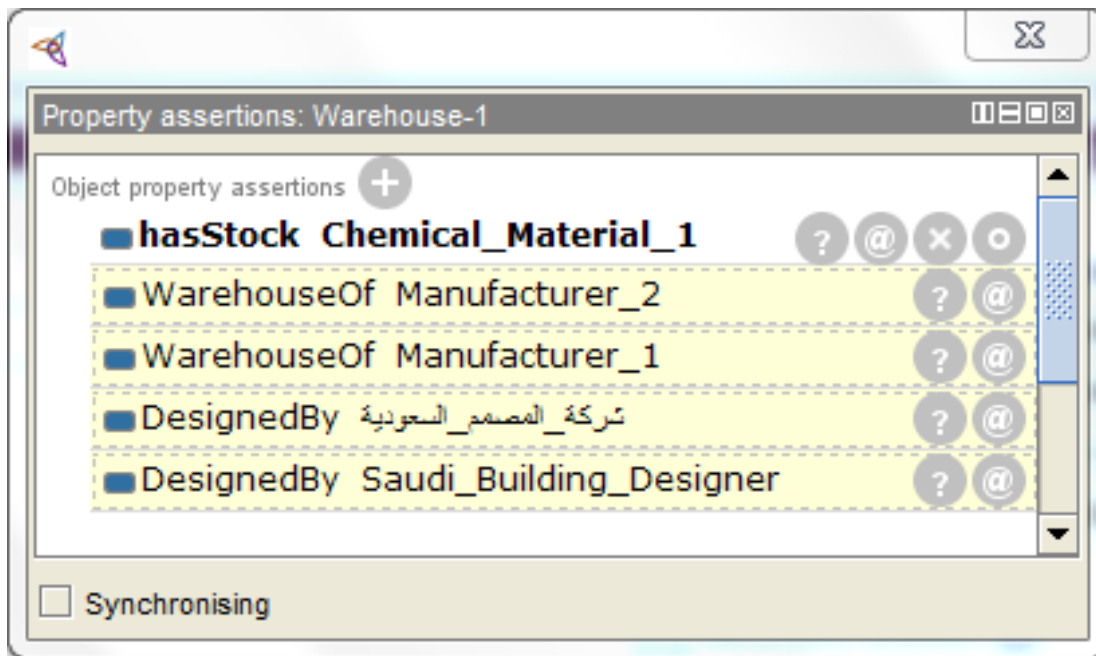
```
<owl:ObjectProperty rdf:about="&scm:#DesignerOf">
  <rdf:type rdf:resource="&owl:InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="&scm:#Company"/>
  <rdfs:range rdf:resource="&scm:#Construction"/>
  <owl:inverseOf rdf:resource="&scm:#DesignedBy"/>
</owl:ObjectProperty>
```





### Inference

If 'Warehouse\_1' is designed by 'Saudi Designers' and 'المصمم شركة السعودية' (Same company name in Arabic Language) then 'المصمم شركة السعودية' and 'Saudi Designers' are same individuals.



### **(e) Reflexive Property**

There is no useful reflexive property identified in the ontology of construction material, however "knows" and "likes" are the example of reflexive properties.

Inference:

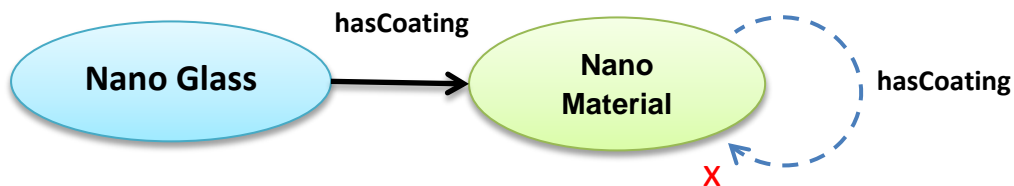
If Person "A" knows person "B", then Person "A" knows himself.

## (f) Irreflexive Property

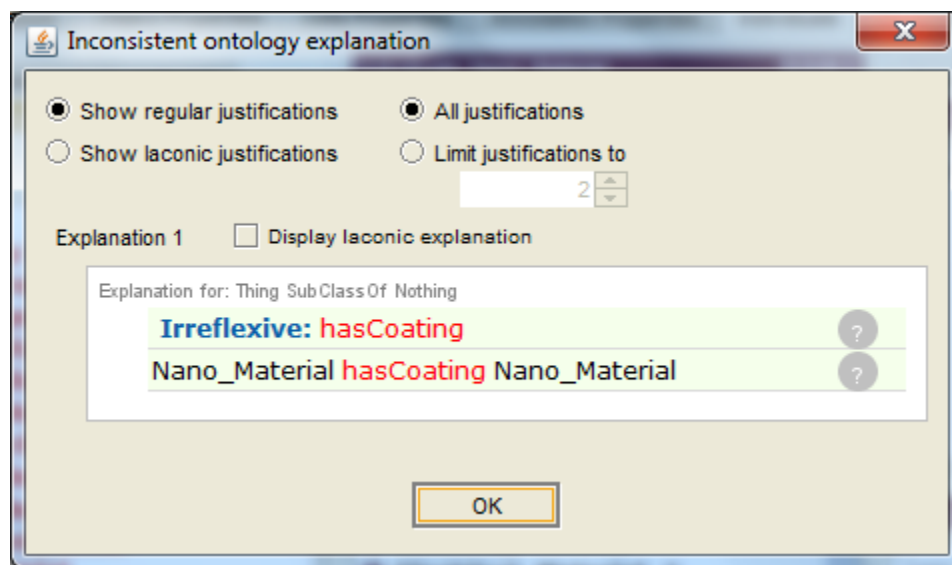
Example: [hasCoating](#)

```
<owl:ObjectProperty rdf:about="&scm:#hasCoating">
  <rdf:type rdf:resource="&owl:IrreflexiveProperty"/>
  <rdfs:domain rdf:resource="&scm:#Material"/>
  <rdfs:range rdf:resource="&scm:#Material"/>
</owl:ObjectProperty>
```

### Inference



If “Nano\_material” is used for Coating, it cant be used for Coating to itself. i.e. a coating material cant be used to coat on itself.



### Conclusion

Thank you Emma and Monika for giving me such wonderful knowledge

### References

- [1] Yi. Hong., Monika Salonki, Lecture Notes, Semantic Web, University of Leicester 2016.
- [2] <http://www.w3.org/standards/semanticweb> [Assessed March 2016)