

# Requirements reasoning using OWL: Tag versus product types

johanw

2017-03-31

## Contents

<b>1</b>	<b>What is this?</b>	<b>2</b>
<b>2</b>	<b>Resources</b>	<b>2</b>
<b>3</b>	<b>The ontology at a glance</b>	<b>2</b>
<b>4</b>	<b>Case description</b>	<b>2</b>
4.1	The Facility . . . . .	2
4.2	A NORSOK U-001 project, and M-650 qualified manufacturers . . . . .	3
4.3	Three valve product types . . . . .	3
4.4	Reasoning about requirements . . . . .	4
4.4.1	Inherited properties . . . . .	4
4.4.2	Test ACME A . . . . .	4
4.4.3	Test ACME B . . . . .	5
4.4.4	Test SuperX . . . . .	5
<b>5</b>	<b>Definitions</b>	<b>6</b>
5.1	Declarations . . . . .	6
5.1.1	Prefixes . . . . .	6
5.1.2	Ontology . . . . .	7
5.2	Classes . . . . .	8
5.2.1	<i>lci:Activity</i> . . . . .	8
5.2.2	<i>lci:PhysicalObject</i> . . . . .	8
5.2.3	<i>lci:Site</i> . . . . .	9
5.2.4	<i>lci:Organisation</i> . . . . .	9
5.3	Object relations (object properties) . . . . .	9
5.3.1	<i>lci:partOf</i> . . . . .	9
5.3.2	<i>lci:connectedTo</i> . . . . .	9
5.3.3	<i>lci:interests</i> . . . . .	9
5.3.4	<i>lci:hasQuality</i> . . . . .	10
5.3.5	<i>lci:qualityQuantifiedAs</i> . . . . .	10
5.3.6	<i>lci:datumUOM</i> . . . . .	10
5.3.7	<i>lci:locatedRelativeTo</i> . . . . .	10
5.3.8	<i>rdf:is_about</i> . . . . .	11
5.3.9	<i>rdf:mentioned_in</i> . . . . .	11
5.4	Data relations (data properties) . . . . .	11
5.4.1	<i>lci:datumValue</i> . . . . .	11

5.4.2	<i>lci:qualityQuantityValue</i>	11
5.5	Annotation relations (annotation properties)	12
5.5.1	<i>foaf:depiction</i>	12
5.5.2	<i>skos:altLabel</i>	12
<b>6</b>	<b>Individuals and pattern classes</b>	<b>12</b>
6.1	Plant, Manifold, and Piping Segment	12
6.2	Valve Tag (the item in focus)	12
6.3	Project	12
6.4	Manufacturers	13
6.5	Valve Product Classes	13

## 1 What is this?

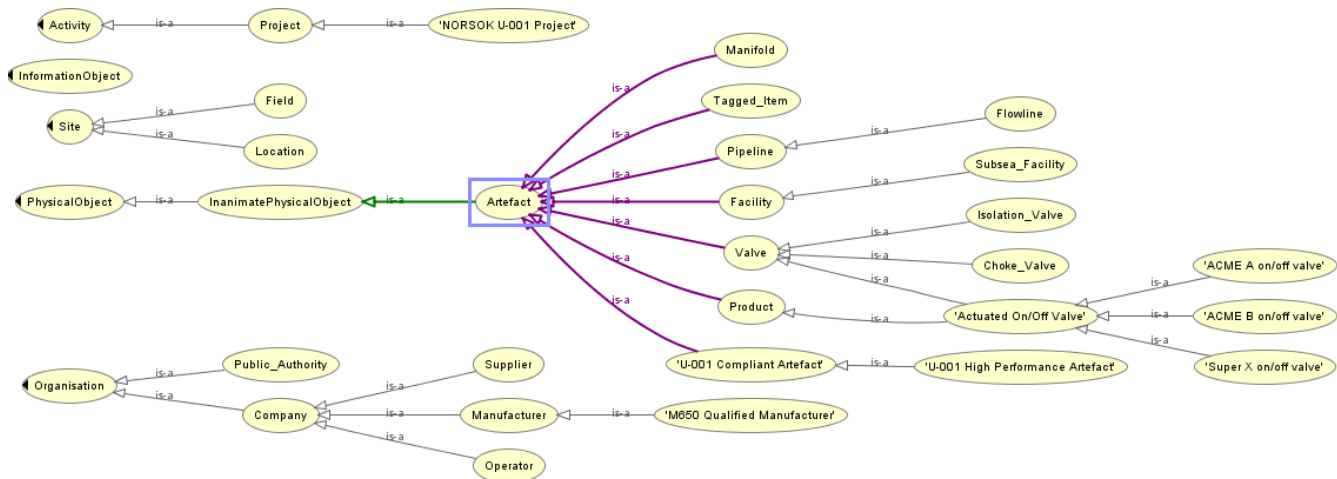
This document defines an example to demonstrate how ontology reasoning can serve to check the requirements on a “functional location” in a plant – a *tag* – versus candidate product types.

The example is loosely based on a highly simplified subsea scenario.

## 2 Resources

- emails from Bjørn Berli (EPIM) and David Cameron (Sirius)
- discussions with Anders Gjerver (Aibel)
- work on ISO 15926 “DL profile”

## 3 The ontology at a glance

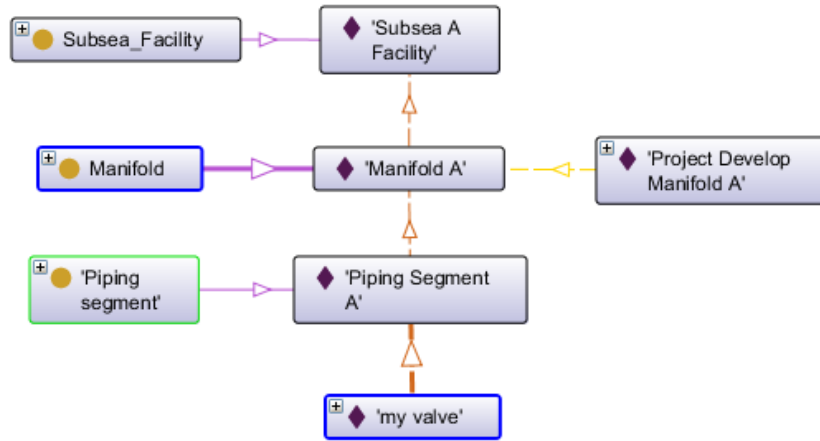


## 4 Case description

### 4.1 The Facility

We have a subsea facility called Subsea A. This has a part Manifold A, which has part Piping Segment A, and again a part *my valve*, which is a tag – a “functional object”.

The project Develop Manifold A is responsible for designing the manifold.

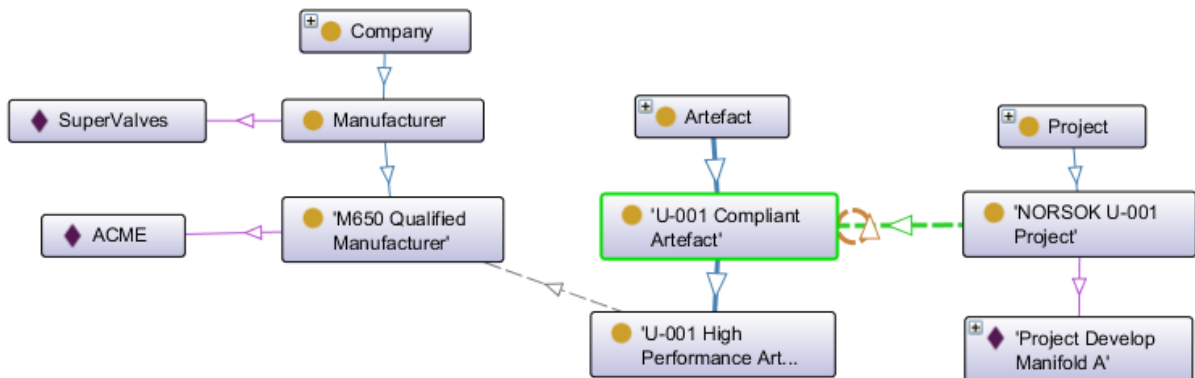


## 4.2 A NORSOK U-001 project, and M-650 qualified manufacturers

The development project is a NORSOK U-001 project. This class of projects carries a constraint saying all designed artefacts are U-001 Compliant.

Furthermore, the ontology states that all Isolation Valves designed *in this particular project* are “High Performance” high alloy items, which carries a requirement (from U-001) that they may only be manufactured by M-650 qualified manufacturers.

There are two manufacturers, ACME and SuperValves. ACME has the M-650 qualification. For SuperValves, we know only that it is a Manufacturer.



## 4.3 Three valve product types

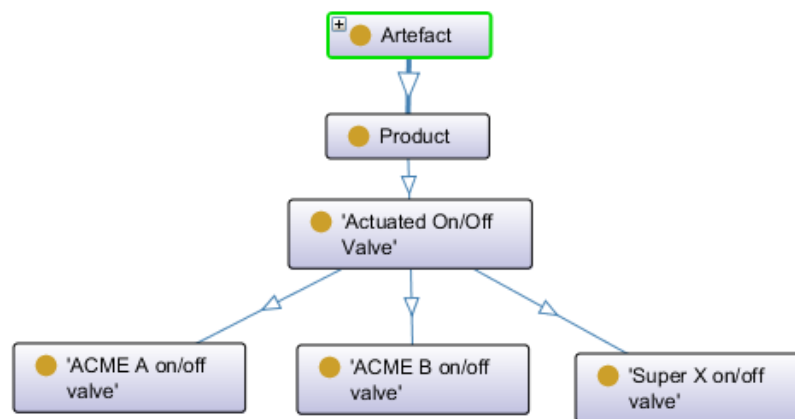
There are three valve *product* classes defined, ACME A, ACME B, and Super X. We have chosen values for some selected attributes so that ACME A will not be suitable for the *my valve* tag, but the other two have sufficient attribute values.

These are the products:

Valve_id	Name	Class	Manufacturer_name	MWP	Cv	Tmin	Tmax
ACME_Valve_A	ACME A on/off valve	rdl:Actuated_OnOff_Valve	ACME	10000	206	-46	170
ACME_Valve_B	ACME B on/off valve	rdl:Actuated_OnOff_Valve	ACME	15000	130	-46	200
Super_Valve_X	Super X on/off valve	rdl:Actuated_OnOff_Valve	SuperValves	17000	240	-46	600

This is *my valve* (Operating Temp will be matched with Tmin, Tmax in the above table).

MWP	Cv	Operating Temp
9000	120	200



## 4.4 Reasoning about requirements

### 4.4.1 Inherited properties

A main point of ontologies is to get *implicit* content out. In the following screenshot from Protégé, inferred attributes are shown on yellow. Running the reasoner on the ontology as-is reveals that *my valve* is a High Performance artefact (left). We also see that the part-whole relationship is followed all the way up to the facility.

Description: 'my valve'	Property assertions: 'my valve'
Types + ● Isolation_Valve ● 'U-001 High Performance Artefact'	Object property assertions + ■ partOf 'Flowline A' ■ partOf 'Subsea A Facility' ■ partOf 'Manifold A'


Now we proceed to test our three classes as products to implement the *my valve* tag.

### 4.4.2 Test ACME A

ACME A has a maximum operating temperature that is too low. When we classify *my valve* as an ACME A,

Description: 'my valve'
Types + ● 'ACME A on/off valve' ● Isolation_Valve ● 'U-001 High Performance Artefact'

we are informed that the ontology is inconsistent.



Your ontology is inconsistent which means that the OWL reasoner will no longer be able to provide any useful information about the ontology.

You have several options at this point:

- Click the Explain button to try the Protege explanation facility.
- If you think you know what the problem is, click Cancel to fix the ontology yourself.
- Some reasoners come with command line tools that will provide complete explanations for inconsistent ontologies.

The reasoner helpfully explains the error as a violation of the temperature requirement – *my valve* requires 200 C, but ACME A has a maximum of 176 C.

```
'my valve' Type 'ACME A on/off valve'
'ACME A on/off valve' SubClassOf valOperatingTemperature_C only xsd:integer[>= -46 , <= 176]
'my valve' valOperatingTemperature_C 200
```

#### 4.4.3 Test ACME B

When we classify *my valve* as ACME B, no inconsistency is reported.

Description: 'my valve'

Property assertions: 'my valve'

Types

- 'ACME B on/off valve'
- Isolation\_Valve
- 'U-001 High Performance Artefact'

Same Individual As

Object property assertions

- partOf 'Flowline A'
- interests ACME
- partOf 'Subsea A Facility'
- partOf 'Manifold A'
- manufacturedBy ACME

This means, **as far as we know, ACME B is an acceptable choice of product for the *my valve* tag.**

#### 4.4.4 Test SuperX

When we classify *my valve* as Super X, no inconsistency is reported here either.

Description: 'my valve'

Property assertions: 'my valve'

Types

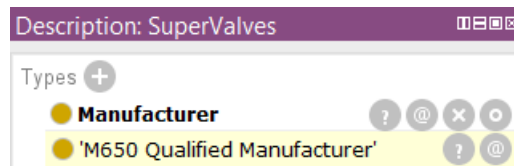
- Isolation\_Valve
- 'Super X on/off valve'
- 'U-001 High Performance Artefact'

Same Individual As

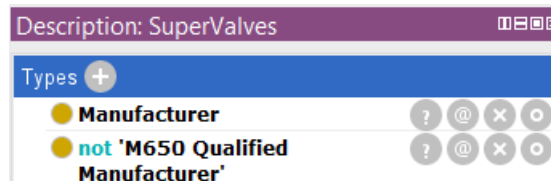
Object property assertions

- partOf 'Flowline A'
- interests SuperValves
- partOf 'Subsea A Facility'
- partOf 'Manifold A'
- manufacturedBy SuperValves

As long as the ontology doesn't explicitly state that SuperValves is not an M-650 qualified manufacturer, it will not infer that SuperX is inadmissible. Indeed, the reasoner has inferred that SuperValves *is* M-650 qualified:



We can however add the plausible statement that SuperValves is not qualified.



Now, the rather complex requirement that all Isolation Valves designed in the current project are (have to be) High Performance items is activated, and we obtain an inconsistency. In actual cases, tracing the inadmissible choice of product back to the violation may be quite difficult. The explanation feature of the ontology reasoner however provides us with a path to the root of the problem.

```
'my valve' Type Isolation_Valve
'my valve' partOf 'Flowline A'
Transitive: partOf
'Project Develop Manifold A' Type designs exactly 0 (hasPart some (Isolation_Valve and (not ('U-001 High Performance Artefact'))))
'Flowline A' partOf 'Manifold A'
'U-001 High Performance Artefact' SubClassOf manufacturedBy only 'M650 Qualified Manufacturer'
'Super X on/off valve' SubClassOf manufacturedBy value SuperValves
hasPart InverseOf partOf
'Project Develop Manifold A' designs 'Manifold A'
SuperValves Type not ('M650 Qualified Manufacturer')
'my valve' Type 'Super X on/off valve'
```

This explanation is rather complex, but it is discovered entirely by following facts that would be given up-front – no coding or ad-hoc rules were needed. Conjecture: even if hundreds of requirements are added to a case model, the explanations are not likely to be much more complex than what we see in this example.

## 5 Definitions

### 5.1 Declarations

#### 5.1.1 Prefixes

The source blocks prefix-block-omn and prefix-block-ttl should have the same contents, in Manchester Syntax and Turtle, respectively.

Manchester syntax:

```
## Prefixes
Prefix: lci: <http://standards.iso.org/iso/15926/>
Prefix: pd: <http://example.org/pd/>
Prefix: rdl: <http://example.org/rdl/>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: xml: <http://www.w3.org/XML/1998/namespace>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

Prefix: skos: <http://www.w3.org/2004/02/skos/core#>  
Prefix: pav: <http://purl.org/pav/>  
Prefix: foaf: <http://xmlns.com/foaf/0.1/>  
Prefix: dcterms: <http://purl.org/dc/terms/>  
Prefix: dcmitype: <http://purl.org/dc/dcmitype/>  
Prefix: npdv: <http://sws.ifi.uio.no/vocab/npd-v2#>  
Prefix: pca: <http://data.posccaesar.org/rdl/>  
Prefix: prov: <http://www.w3.org/ns/prov#>

#### Turtle:

```
## Prefixes
@prefix lci: <http://standards.iso.org/iso/15926/> .
# note, not same as the the namespace of the CD, which looks inappropriate .
@prefix pd: <http://example.org/pd/> .
@prefix rdl: <http://example.org/rdl/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix pav: <http://purl.org/pav/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dcmitype: <http://purl.org/dc/dcmitype/> .
@prefix npdv: <http://sws.ifi.uio.no/vocab/npd-v2> .
@prefix pca: <http://data.posccaesar.org/rdl/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
```

### 5.1.2 Ontology

#### core

```
Ontology: <http://example.org/req-concept>
Annotations: rdfs:label "Requirements demonstration",
              owl:versionInfo "Started 2017-03-30",
              rdfs:comment "Demonstration ontology showing tags"
Import: <http://standards.iso.org/iso/15926/-12/tech/ontology/DL-profile>
```

## 5.2 Classes

## Class details

### 5.2.1 *Ici:Activity*

. ***rdl:Project***

. . ***NORSOK U-001 Project (rdl:U001\_Project)***

Class: rdl:U001\_Project

SubClassOf: rdl:designsArtefact only rdl:U001\_Compliant\_Artefact

### 5.2.2 *Ici:PhysicalObject*

. ***Ici:InanimatePhysicalObject***

. . ***rdl:Artefact***

. . . ***rdl:Pipeline***

. . . . ***rdl:Flowline***

. . . ***Piping segment (rdl:Piping\_Segment)***

. . . ***rdl:Valve***

. . . . ***rdl:Choke\_Valve***

. . . . ***rdl:Isolation\_Valve***

. . . ***rdl:Manifold***

. . . ***rdl:Product***

. . . . ***Actuated On/Off Valve (rdl:Actuated\_OnOff\_Valve)***

Class: rdl:Actuated\_OnOff\_Valve

SubClassOf: rdl:Valve

. . . ***rdl:Facility***

. . . . ***rdl:Subsea\_Facility***

. . . ***rdl:Tagged\_Item***

*rdfs:comment* Tagged items are typically replaceable units in an assembly.

. . . ***U-001 Compliant Artefact (rdl:U001\_Compliant\_Artefact)***

*rdfs:comment* Class of artefacts compliant with NORSOK U-001, “Subsea production systems”

Class: rdl:U001\_Compliant\_Artefact

SubClassOf: Ici:hasPart only rdl:U001\_Compliant\_Artefact

. . . . ***U-001 High Performance Artefact (rdl:U001\_HighP\_Artefact)***

*rdfs:comment* Class of U-001 artefacts to match NORSOK U-001 paragraph 5.14.1: “Manufacturers of components in special materials such as 22Cr and 25Cr Duplex stainless steels, 6Mo and other high alloy stainless steels, nickel alloys and titanium castings shall be qualified in accordance with NORSOK M-650.”



Class: rdl:U001\_HighP\_Artefact  
SubClassOf: rdl:manufacturedBy only rdl:M650\_Qualified

### 5.2.3 Ici:Site

. **rdl:Location**

. **rdl:Field**

### 5.2.4 Ici:Organisation

. **rdl:Company**

. . **rdl:Operator**

. . **rdl:Manufacturer**

. . . **M650 Qualified Manufacturer (rdl:M650\_Qualified)**

*rdfs:comment* See NORSOK U-001, 5.14.1

. . **rdl:Supplier**

*skos:altLabel* Vendor

. **rdl:Public\_Authority**

## 5.3 Object relations (object properties)

## Object property details

### 5.3.1 Ici:partOf

*rdfs:comment* For this example, we make the part-of relation transitive.

ObjectProperty: Ici:partOf  
Characteristics: Transitive

### 5.3.2 Ici:connectedTo

. **Ici:directlyConnectedTo**

### 5.3.3 Ici:interests

. **designs (rdl:designsArtefact)**

*rdfs:comment* Relation from a project to artefacts designed in the project.

ObjectProperty: rdl:designsArtefact  
Domain: Ici:Activity  
Range: rdl:Artefact

**. *manufacturedBy* (rdl:manufacturedBy)**

#### **5.3.4 *lci:hasQuality***

**. *rdl:hasPhysicalQuantity***

**. . *rdl:hasMass***

**. . *rdl:hasPressure***

**. . *rdl:hasTemperature***

#### **5.3.5 *lci:qualityQuantifiedAs***

**. *rdl:qualityMeasuredAs***

#### **5.3.6 *lci:datumUOM***

#### **5.3.7 *lci:locatedRelativeTo***

**. *rdl:resides\_in***

*rdfs:comment* This relation is for stating that a physical object is located in a Site. Note that this is distinct from one object being contained by another – for which, see *containedBy*.

ObjectProperty: rdl:resides\_in  
Domain: lci:PhysicalObject  
Range: lci:Site

**. *rdl:has\_resident***

*rdfs:comment* The inverse of “resides in”, this relates a site (e.g., an area) to items that are situated in the area.

ObjectProperty: rdl:has\_resident  
InverseOf: rdl:resides\_in

**. *rdl:holds\_location***

*rdfs:comment* Relation to state that a physical object holds a location. For example, platform EG (a facility) holds location H00; or a fuse box has locations for fuses.

ObjectProperty: rdl:holds\_location  
Domain: lci:PhysicalObject  
Range: rdl:Location # Location

**. *rdl:is\_location\_in\_physical\_object***

*rdfs:comment* The inverse relation of “holds location”, a relation to state that a location is defined by a physical object, like a process facility or an electrical equipment cabinet.

ObjectProperty: rdl:is\_location\_in\_physical\_object  
InverseOf: rdl:holds\_location

### **. rdl:has\_sub-location**

*rdfs:comment* Inverse of “is sub-location of”

ObjectProperty: rdl:has\_sub-location  
InverseOf: rdl:is\_sub-location\_of

### **. rdl:is\_sub-location\_of**

*rdfs:comment* Relation for sub-locations; cf. “resides in” for devices placed in locations.

ObjectProperty: rdl:is\_sub-location\_of  
Domain: rdl:Location #Location  
Range: rdl:Location #Location  
Characteristics: Transitive

## **5.3.8 rdl:is\_about**

*rdfs:seeAlso* [http://purl.obolibrary.org/obo/IAO\\_0000136](http://purl.obolibrary.org/obo/IAO_0000136)

*rdfs:comment* Following the Information Artifact Ontology, the relation “is about” is a maximally generic relation from information objects to the things they define, describe, mention, or characterise.

ObjectProperty: rdl:is\_about  
Domain: lci:InformationObject

## **5.3.9 rdl:mentioned\_in**

*rdfs:comment* A maximally general relation from a thing to a document that is about the thing; the inverse of “is about”.

ObjectProperty: rdl:mentioned\_in  
InverseOf: rdl:is\_about  
Range: lci:InformationObject

### **. rdl:defined\_in**

*rdfs:comment* Pointer to the document (individual) in which something is defined.

## **5.4 Data relations (data properties)**

## Data property details

### **5.4.1 lci:datumValue**

### **5.4.2 lci:qualityQuantityValue**

DataProperty: lci:qualityQuantityValue  
Domain: lci:PhysicalObject

*rdfs:comment* This is a super-property for “template” relations that combine a quality, the weak lci:qualityQuantifiedAs, and a unit of measure into a simple data property. For instance, “mass in kilograms” can be introduced as such a data property, for expressing the mass of an entity on the kilogram scale. lci:qualityQuantifiedAs is “weak” in the sense that it doesn’t distinguish between designed or estimated, and measured, values.

**. rdl:valDepth\_m**

**. rdl:valNominalSize\_in**

*rdfs:comment* Seat nominal diameter

**. rdl:valMWP\_psi**

**. rdl:valCv**

*rdfs:comment* Common UoM for Cv?

**. rdl:valOperatingTemperature\_C**

## 5.5 Annotation relations (annotation properties)

## Annotation property details

### 5.5.1 foaf:depiction

*rdfs:comment* This FOAF annotation property is for providing illustrations. In the current context, the main interest is in showing modelling diagrams. With a valid image URL, a thumbnail will be displayed in the Protégé editor.

### 5.5.2 skos:altLabel

## 6 Individuals and pattern classes

##

## Individuals (Ontology patterns)

##

### 6.1 Plant, Manifold, and Piping Segment

Individual: pd:subseaA

Annotations: rdfs:label "Subsea A Facility"

Types: rdl:Subsea\_Facility

Facts: rdl:valDepth\_m 900

Individual: pd:manifoldA

Annotations: rdfs:label "Manifold A"

Types: rdl:Manifold

Facts: lci:partOf pd:subseaA

Individual: pd:piping\_segmentA

Annotations: rdfs:label "Piping Segment A"

Types: rdl:Piping\_Segment

Facts: lci:partOf pd:manifoldA, rdl:valNominalSize\_in 4

### 6.2 Valve Tag (the item in focus)

Individual: pd:myValveTag

Annotations: rdfs:label "my valve"

Types: rdl:Isolation\_Valve

Facts: lci:partOf pd:piping\_segmentA,

rdl:valOperatingTemperature\_C 200, rdl:valCv 120, rdl:valMWP\_psi 9000

### 6.3 Project

The project designs the Manifold in question. It also applies a rule that any Isolation Valve shall be a “high performance” artefact. This will imply that any Isolation Valve is made by an M-650 qualified manufacturer.

Individual: pd:projectX  
 Annotations: rdfs:label "Project Develop Manifold A"  
 Types: rdl:U001\_Project,  
       rdl:designsArtefact exactly 0 (lci:hasPart some (rdl:Isolation\_Valve and not rdl:U001\_HighP\_Artefact))  
 Facts: rdl:designsArtefact pd:manifoldA

## 6.4 Manufacturers

Manufacturer_name	Class
ACME	rdl:M650_Qualified
SuperValves	rdl:Manufacturer

Individual: pd:Manufacturer\_ACME  
 Annotations: rdfs:label "ACME"  
 Types: rdl:M650\_Qualified

Individual: pd:Manufacturer\_SuperValves  
 Annotations: rdfs:label "SuperValves"  
 Types: rdl:Manufacturer

## 6.5 Valve Product Classes

Valve_id	Name	Class	Manufacturer_name	MWP	Cv	Tmin	Tmax	Dim
ACME_Valve_A	ACME A on/off valve	rdl:Actuated_OnOff_Valve	ACME	10000	206	-46	176	4
ACME_Valve_B	ACME B on/off valve	rdl:Actuated_OnOff_Valve	ACME	15000	130	-46	204	4
Super_Valve_X	Super X on/off valve	rdl:Actuated_OnOff_Valve	SuperValves	17000	240	-46	600	4

We let the temperature ranges be integers.

Class: pd:ACME\_Valve\_A  
 Annotations: rdfs:label "ACME A on/off valve"  
 SubClassOf: rdl:Actuated\_OnOff\_Valve,  
       rdl:manufacturedBy value pd:Manufacturer\_ACME,  
       rdl:valMWP\_psi only xsd:integer[>=0, <= 10000],  
       rdl:valCv only xsd:integer[>=0, <= 206],  
       rdl:valOperatingTemperature\_C only xsd:integer[>=-46, <=176],  
       rdl:valNominalSize\_in value 4

Class: pd:ACME\_Valve\_B  
 Annotations: rdfs:label "ACME B on/off valve"  
 SubClassOf: rdl:Actuated\_OnOff\_Valve,  
       rdl:manufacturedBy value pd:Manufacturer\_ACME,  
       rdl:valMWP\_psi only xsd:integer[>=0, <= 15000],  
       rdl:valCv only xsd:integer[>=0, <= 130],  
       rdl:valOperatingTemperature\_C only xsd:integer[>=-46, <=204],  
       rdl:valNominalSize\_in value 4

Class: pd:Super\_Valve\_X  
 Annotations: rdfs:label "Super X on/off valve"  
 SubClassOf: rdl:Actuated\_OnOff\_Valve,  
       rdl:manufacturedBy value pd:Manufacturer\_SuperValves,  
       rdl:valMWP\_psi only xsd:integer[>=0, <= 17000],  
       rdl:valCv only xsd:integer[>=0, <= 240],  
       rdl:valOperatingTemperature\_C only xsd:integer[>=-46, <=600],  
       rdl:valNominalSize\_in value 4