Advanced ontologies and reasoning

María Poveda Villalón, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

[mpoveda@fi.upm.es] @MariaPovedaV

SSoLDAC23

Licence

- This work is licensed under Creative Commons Attribution – Non Commercial – Share Alike License
- *You are free:*
  - *to Share — to copy, distribute and transmit the work*
  - *to Remix — to adapt the work*

- Under the following conditions
  - *Attribution — You must attribute the work by inserting*
    - "[source http://www.oeg-upm.net/]" at every reused slide
    - A slide declaring: "This material is partially based on "Ontology Development" by María Poveda Villalón and Alba Fernández Izquierdo"
  - *Non-commercial*
  - *Share-Alike*

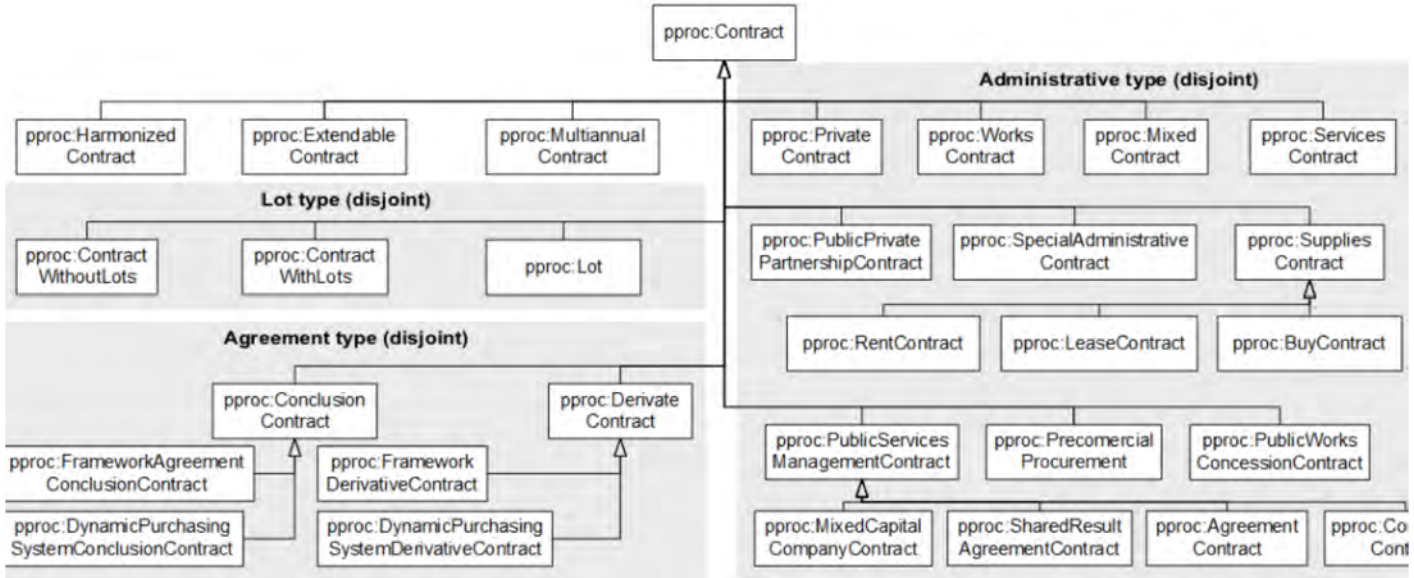
About this material

- The materials for this session has been elaborated by María Poveda Villalón reusing content generated by the following OEG colleagues:
  - Oscar corcho
  - Raúl García-Castro
  - Alba Fernández-Izquierdo
  - Etc.

A vocabulary, built with some consensus, and described formally

Define terms, how they are classified, their properties, relations, etc.

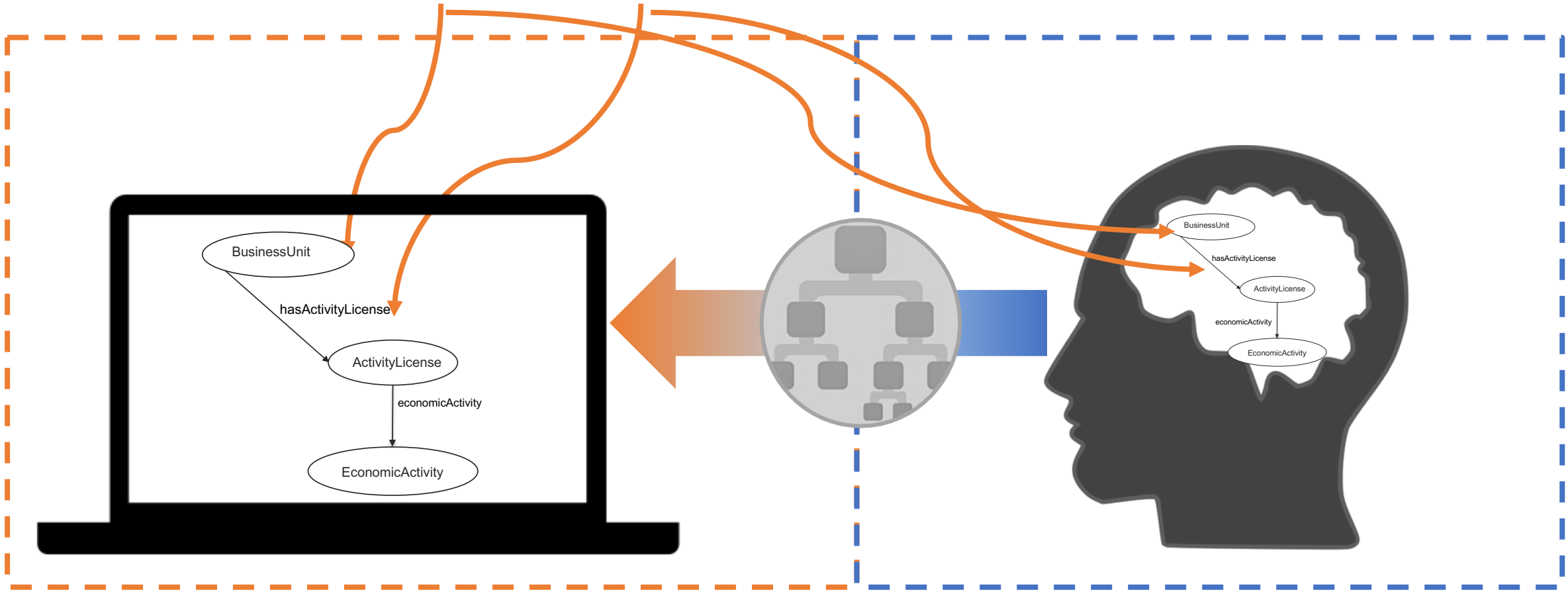
In W3C recommendations (languages) such as RDF Schema or the Web Ontology Language (OWL)

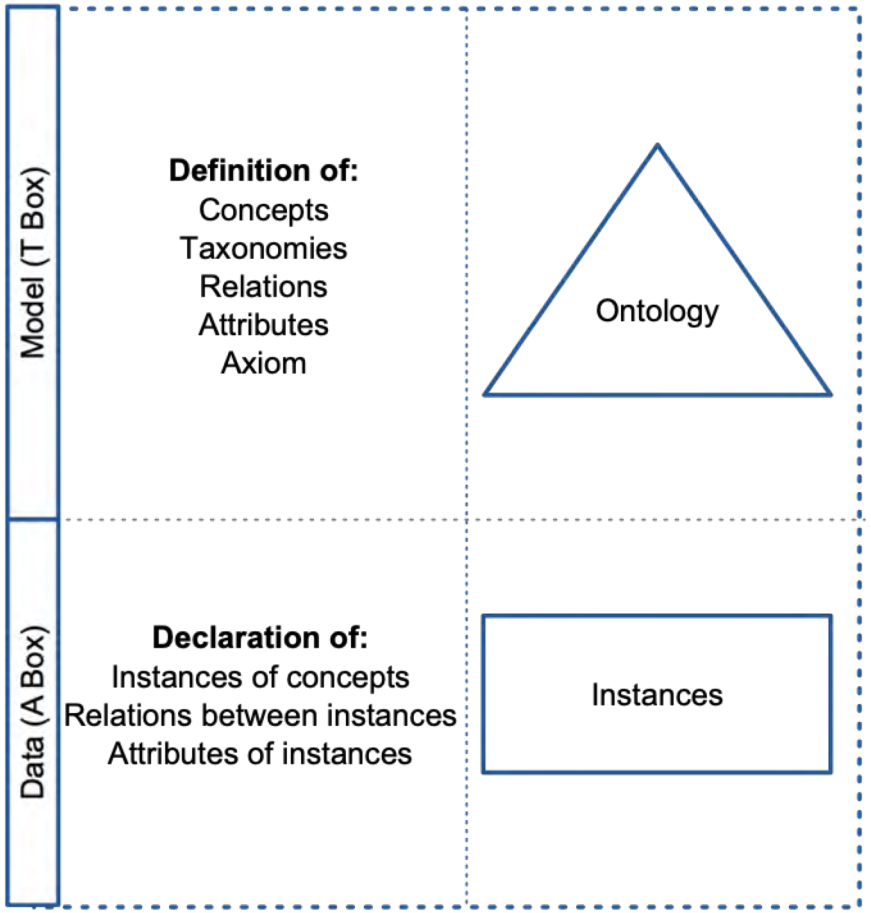


Slide taken from Oscar Corcho

(https://www.w3.org/standards/semanticweb/ontology)

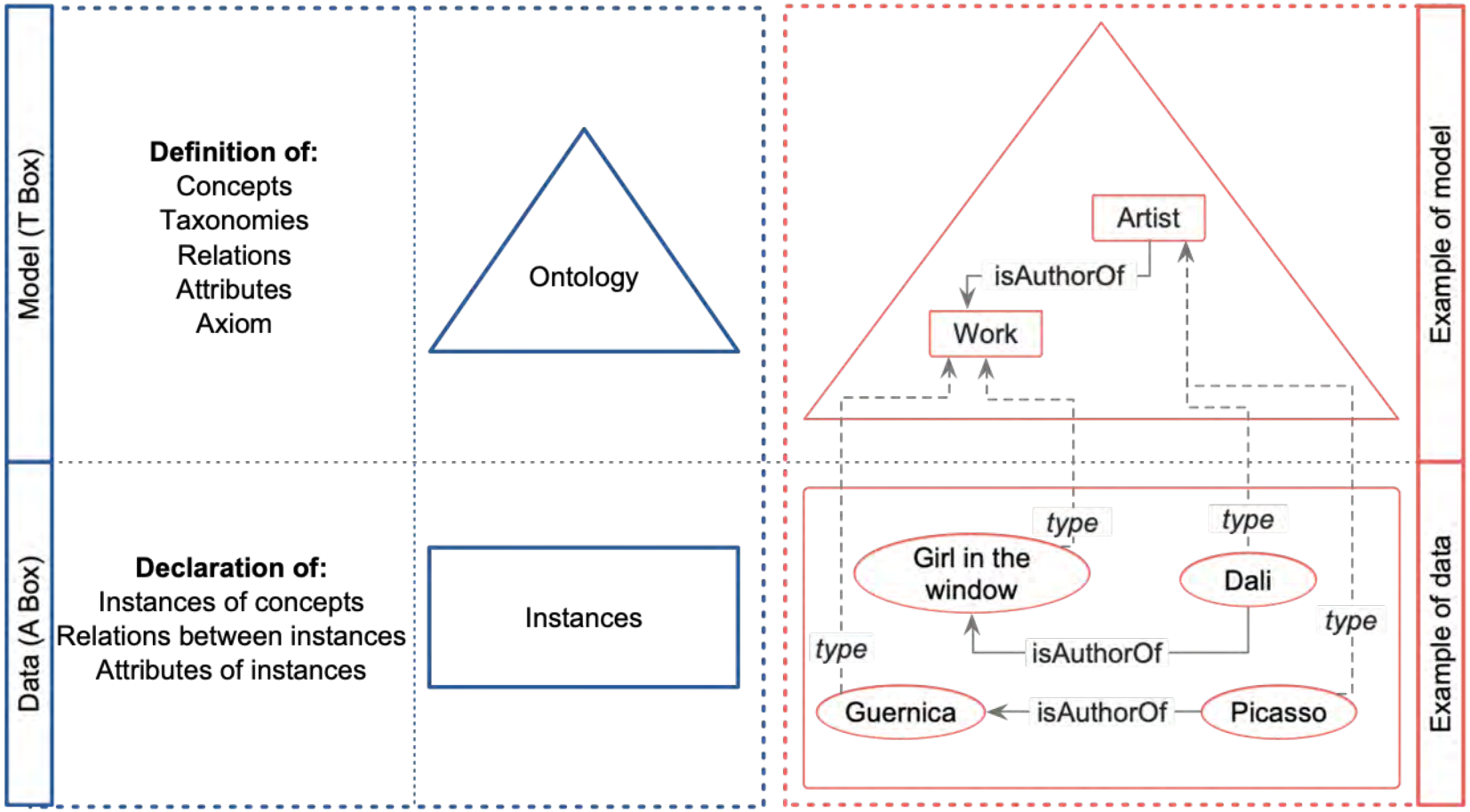
A vocabulary defines the **concepts** and **relations** used to describe and represent a **domain** of interest



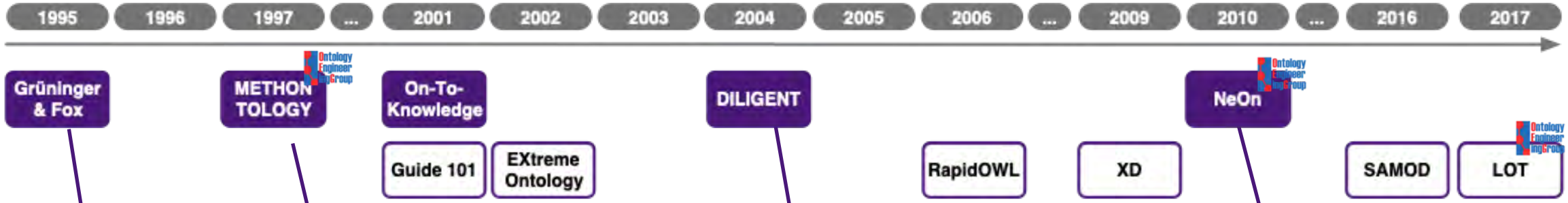


- A knowledge base is divided into:
  - **Tbox** (*Terminological KB*): set of axioms that define the domain:
    - Artist⊏ Person
    - Person ≡ Man ⊔ Woman
  - **Abox** (*Assertional KB*): set of axioms that descibe a situation:
    - Picasso: Man
    - isAuthorOf (Picasso, Guernica)

Structure of *DL* ontologies



Some Ontology Development Methodologies



- Competency Questions technique

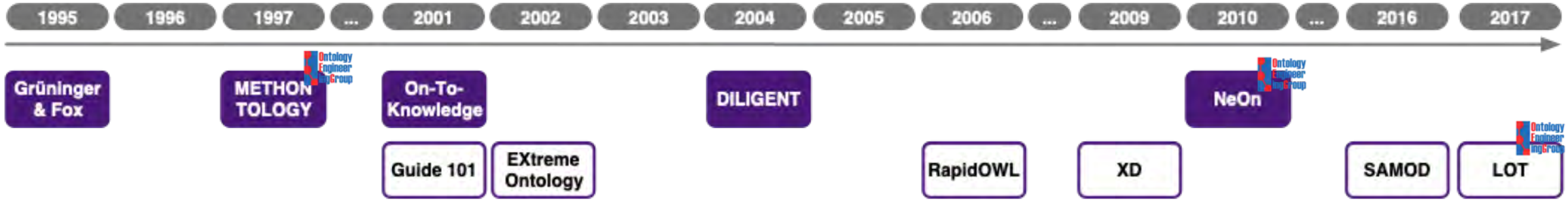
- Intermediate representations
- Activities identification and definition
- Activities organization

- Best practices
- Lightweight
- Common problems addressed
- Frames oriented… but still applicable

- Collaborative process
- Local adaptations

- 9 Scenarios
- Activity templates
- Guidelines for: specification, ontology reuse, NOR reuse, ODP reuse, scheduling, etc.

Ontology Development Methodologies

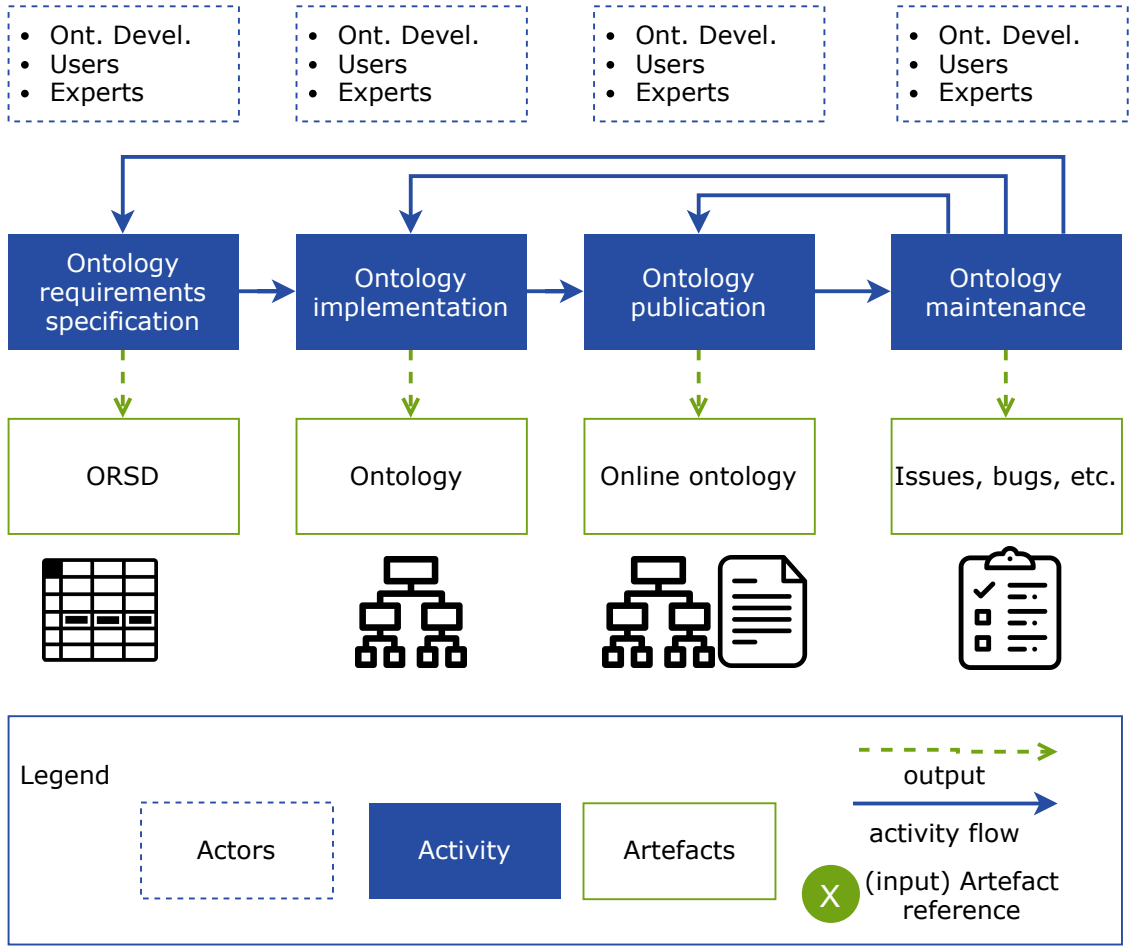


- Towards lightweight and agile processes
- Inspiration from software development practices
- Coupling Software and ontology development

O. Development Methodologies | **Ontology Development Lightweight Approaches**

Slide taken from “Trendy Practices and Tools in Ontological Engineering” by Dr. María Poveda Villalón

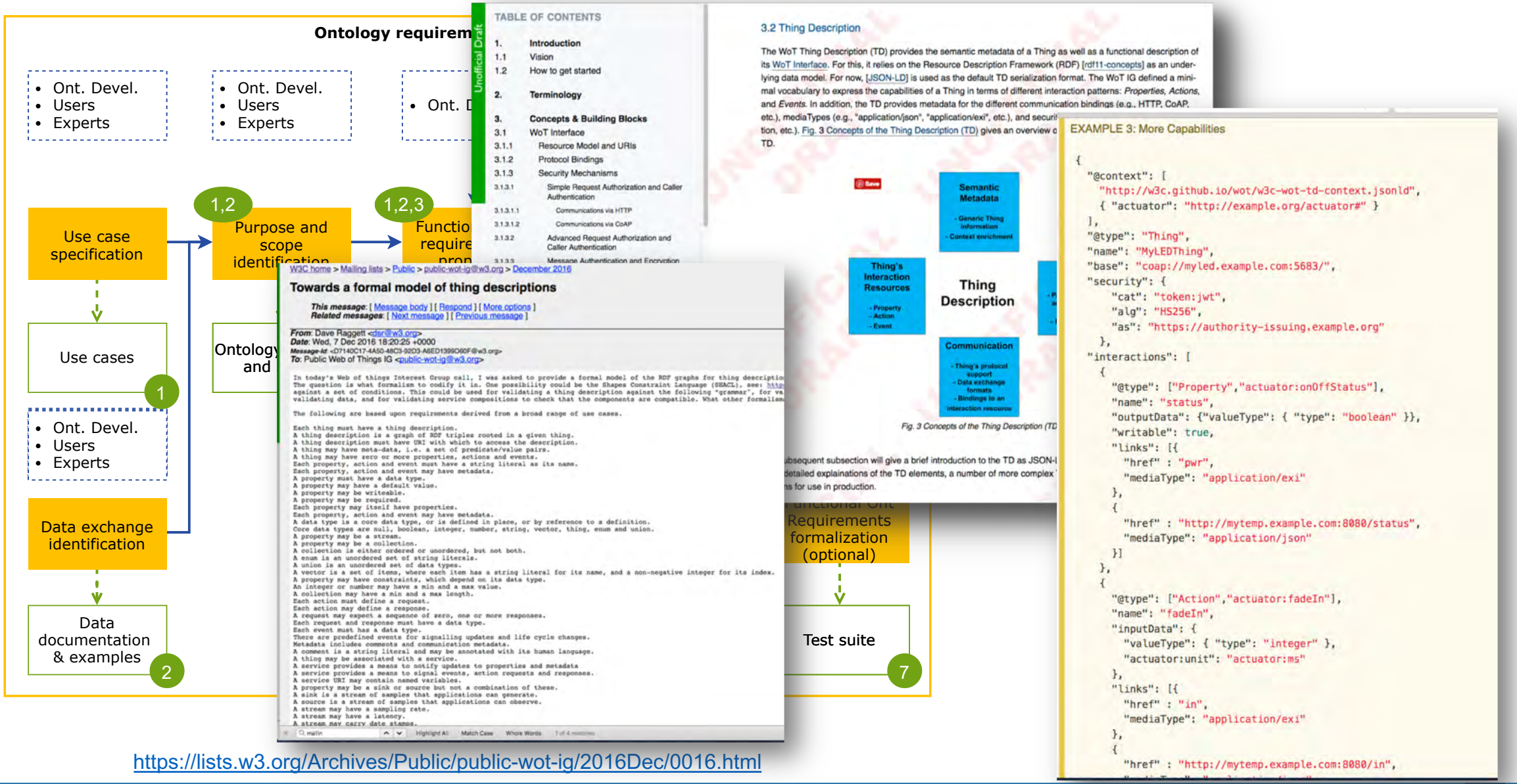
Ontology development process overview



http://lot.linkeddata.es/ https://doi.org/10.1016/j.engappai.2022.104755

Requirement specification

http://w3c.github.io/wot/current-practices/wot-practices



https://lists.w3.org/Archives/Public/public-wot-ig/2016Dec/0016.html

- Following METHONTOLOGY

Confluence Spaces People Create ···

Concept list

Optional: language tags, English by default

Concept	Other names/ids	Description
Building		Building where the behavior occurs
Occupant		Occupants of the buildings
Behavior		Behaviors of the occupants
Space		Internal space of the building
Meeting		Meeting information if the space is communal
System		The system the occupant interact with (Windows, HVAC's,

Confluence Spaces People Create ··· Search

Concept	Attribute	Description	Value type expected (integer, boolean, float, list of specific values, etc.)	Max cardinality	Ordering Needed (applicable for concepts with Max cardinality over 1)	Unit of measure (applicable only for "measure" data types)	Sensitive data (if applicable, e.g. personal data that need to be anonymized in BIF)	Timezone (applicable only for datetime data types)	Related standard (if applicabl
Space	description	Description of the space	String	1					obXML
Space	maxNumberOccupants	Maximum number of occupants	Integer	1					obXML
Space	minNumberOccupants	Minimum number of occupants	Integer	1					obXML
Meeting	meetingDuration	Duration of meeting	Integer	1		seconds			obXML

Confluence Spaces People Create ···

Relations: relations from objects/entities to objects/entities

Concept	Relation	Description	Target object (should appear in the "Concept list")	Max cardinality	Ordering Needed (applicable for concepts with Max cardinality over 1)	(ignore if not sure or not applicable) Other characteristics: symmetric, transitive, has some special behaviour or meaning? etc.	Needed by	standards
Building	hasSpace		Space					
Space	meeting		Meeting					
Space	hasSystem		System					
Space	usedBy		Occupant					

Gómez-Pérez, A., Corcho, O., and Fernández-López, M. (2004). Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Advanced Information and Knowledge Processing. Springer.

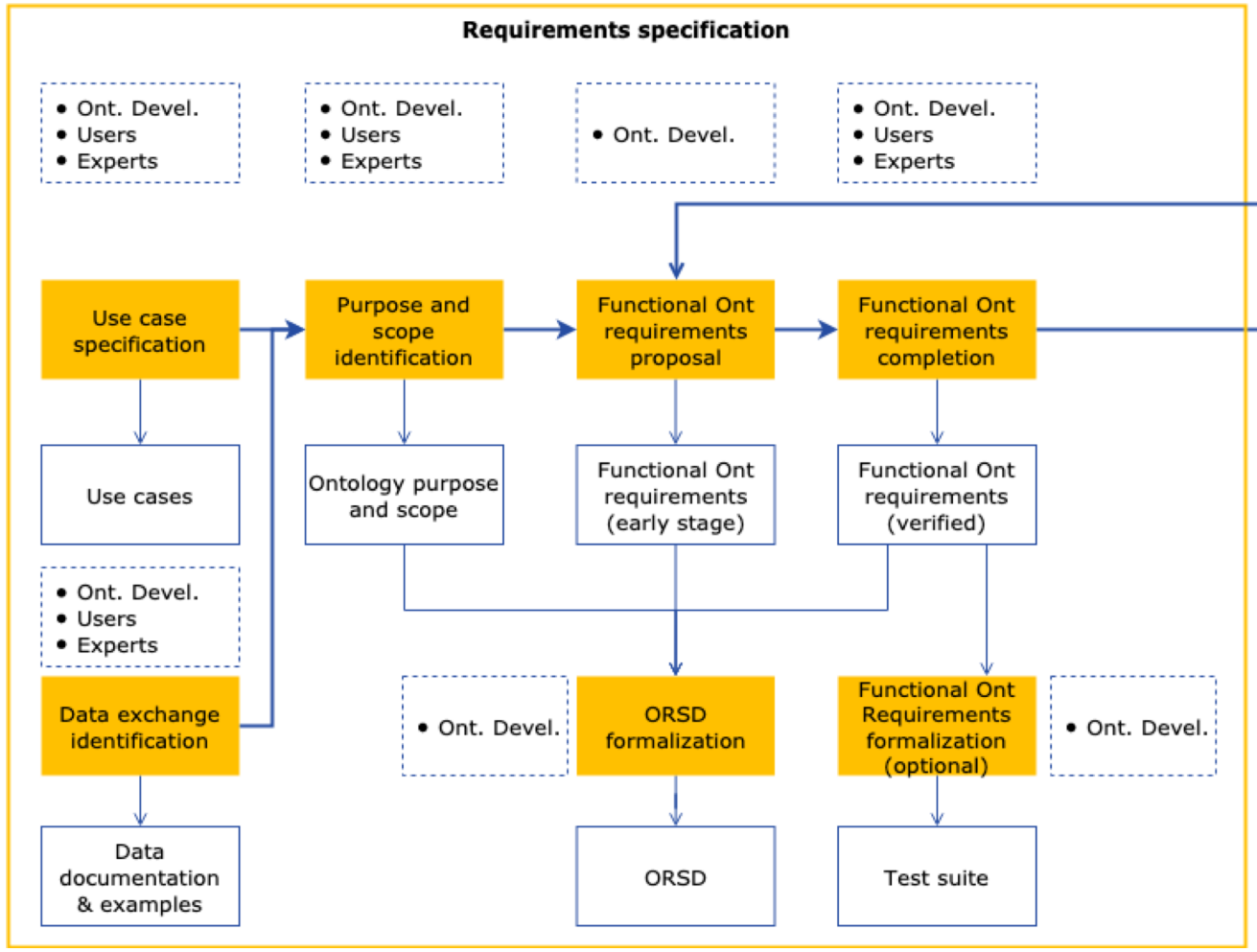
- Following Competency Questions technique

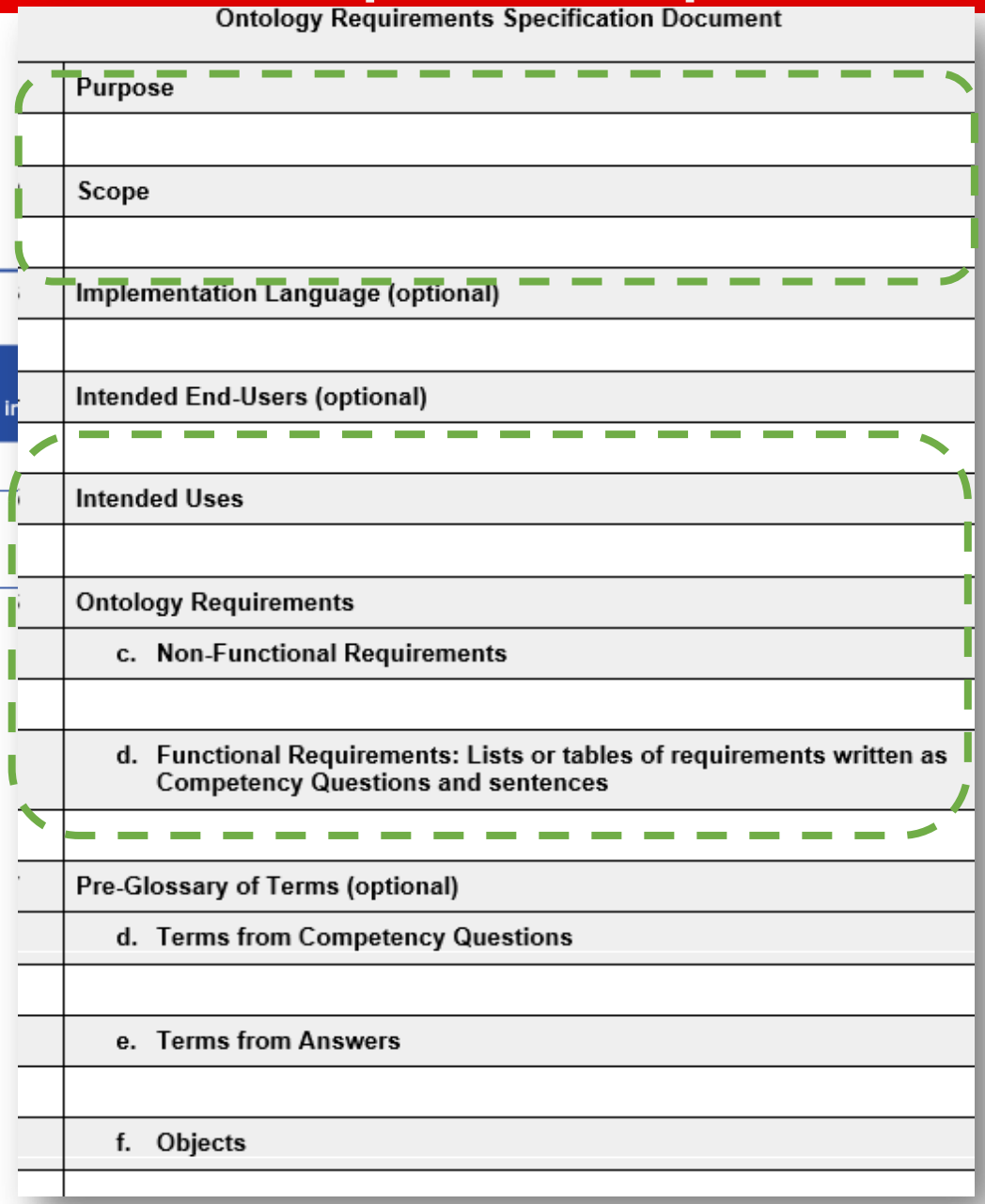
Scope: to support the registration, discovery and search of devices in IoT infrastruc						
Requirement						
Identifier (domain+id)	Sprint	Competency Question / Natural language sentence (fact)	Answer	Status (Proposed, Accepted, Rejected, Deprecated)	Comments	Extracted from (provenance)
WoT1	1	What is a thing in the web thing context?	The abstract concept of a physical entity that can either be a real-world arteact, such as a device, or a virtual entity that represents physicality, such as a room or group of devices	A	In vicinity the things can be sensors and devices but also added-value services.	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT2		What is a servient?	The addressable application endpoint of a Thing that makes it interactive by providing a WoT Interface and means to execute application logic			http://w3c.github.io/wot/current-practices/wot-practices.html
WoT3		What is a repository?	A registry for Thing Descriptions that provides a Web interface to register Thing Descriptions and look them up, for intance using SPARQL queries			http://w3c.github.io/wot/current-practices/wot-practices.html
WoT4	-	What is a thing description?	An RDF document (currently serialized in JSON-LD by default) that contains semantic and functional descriptions of a Thing	R	covered in core	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT5	-	What is a WoT interface?	Def 1: Resource-oriented Web interface (often called "Web API") that allows access to servients over the network using different Protocol Bindings. Def 2: A WoT interface is also a web API that follows the recommendations of the WoT Interest Group.	R	https://github.com/mariapoveda/vicinity-ontology-wot/issues/5 Obsolete:Añadir a la descripción de WoT Interface. Para que un api web sea una WoT interface tiene que ser describle por un TD. It has to have at least one property.	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT6		Things in the WoT architecture are represented by servients,				http://w3c.github.io/wot/current-practices/wot-practices.html
WoT7		Servient can represent virtual things as well.			How are the hosts represented? URIs? entities? strings?	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT8		Servients are hosted anywhere, a smartphone, local gateway, or the cloud.				http://w3c.github.io/wot/current-practices/wot-practices.html
WoT9		Servient communicate to each other through a WoT interface				http://w3c.github.io/wot/current-practices/wot-practices.html
WoT10		Servient can have cliente or server roles or both.			Is this fixed or can change? Needs to be tracked? Need to be attached to time intervals?	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT11	-	Each thing is described by WoT Thing Descriptions.	A thing might be described by more than one thing description	R	https://github.com/mariapoveda/vicinity-ontology-wot/issues/1 (covered in core)	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT12		Thing Descriptions can be registered in Td repositories.			Which attributes or waht information should be attached to tha TD repository?	http://w3c.github.io/wot/current-practices/wot-practices.html
WoT13	-	TD repositories can be queried using for example SPARQL		R		http://w3c.github.io/wot/current-practices/wot-practices.html
WoT14	-	A WoT interface provides Web resources that implement the interaction patterns of Properties, Actions, and Events. Each web resource acts as a mediator/proxy (web interface) between the Thing and their clients		R	https://github.com/mariapoveda/vicinity-ontology-wot/issues/5	http://w3c.github.io/wot/current-practices/wot-practices.html

Shared in **online** spreadsheets

Grüninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. In IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing

Requirement specification



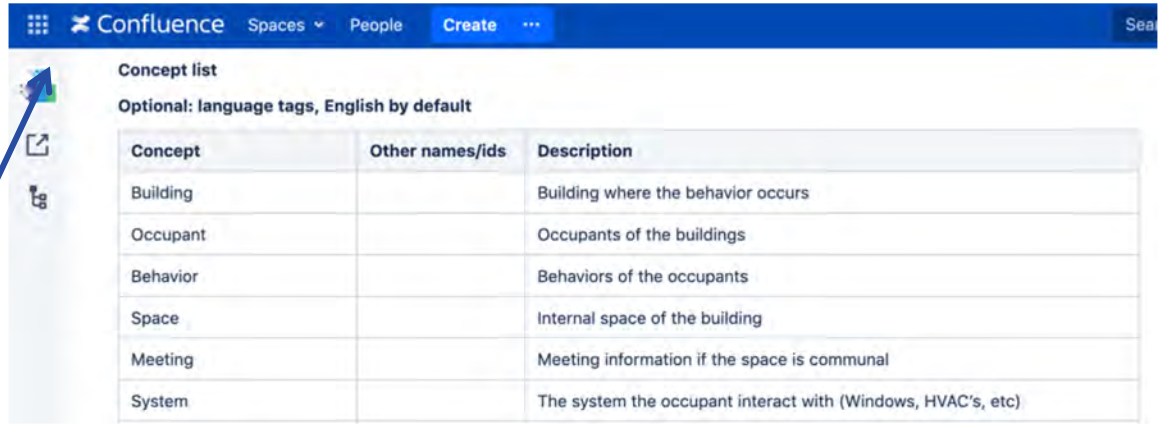


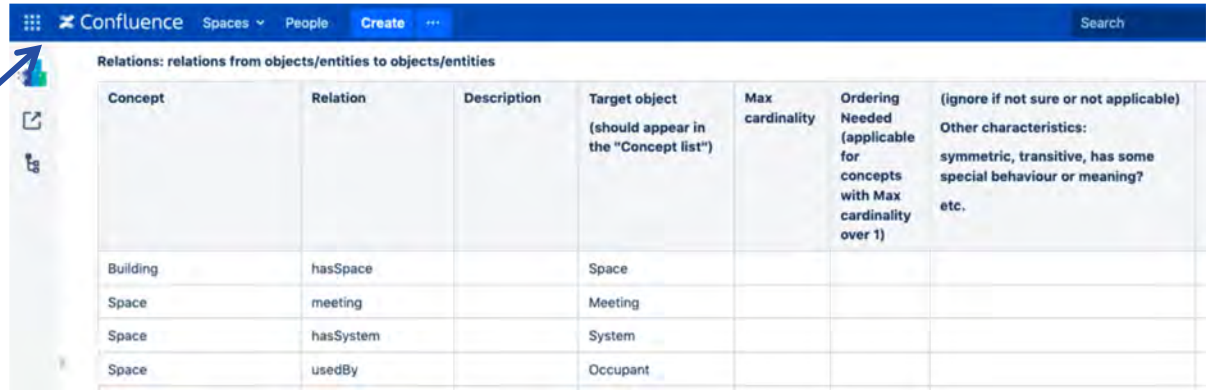
CORAL corpus provides examples and requirements patterns

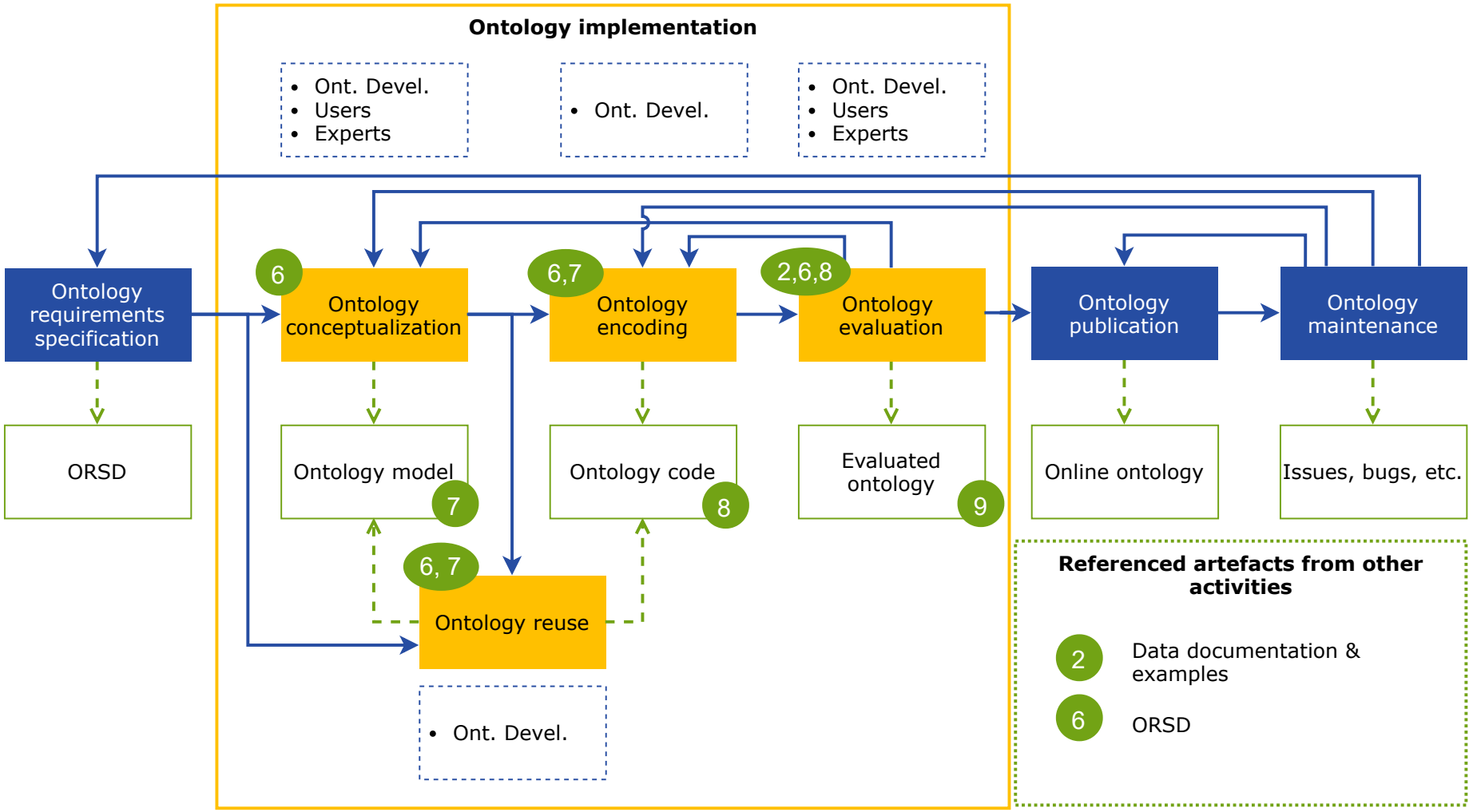
http://coralcorpus.linkeddata.es/

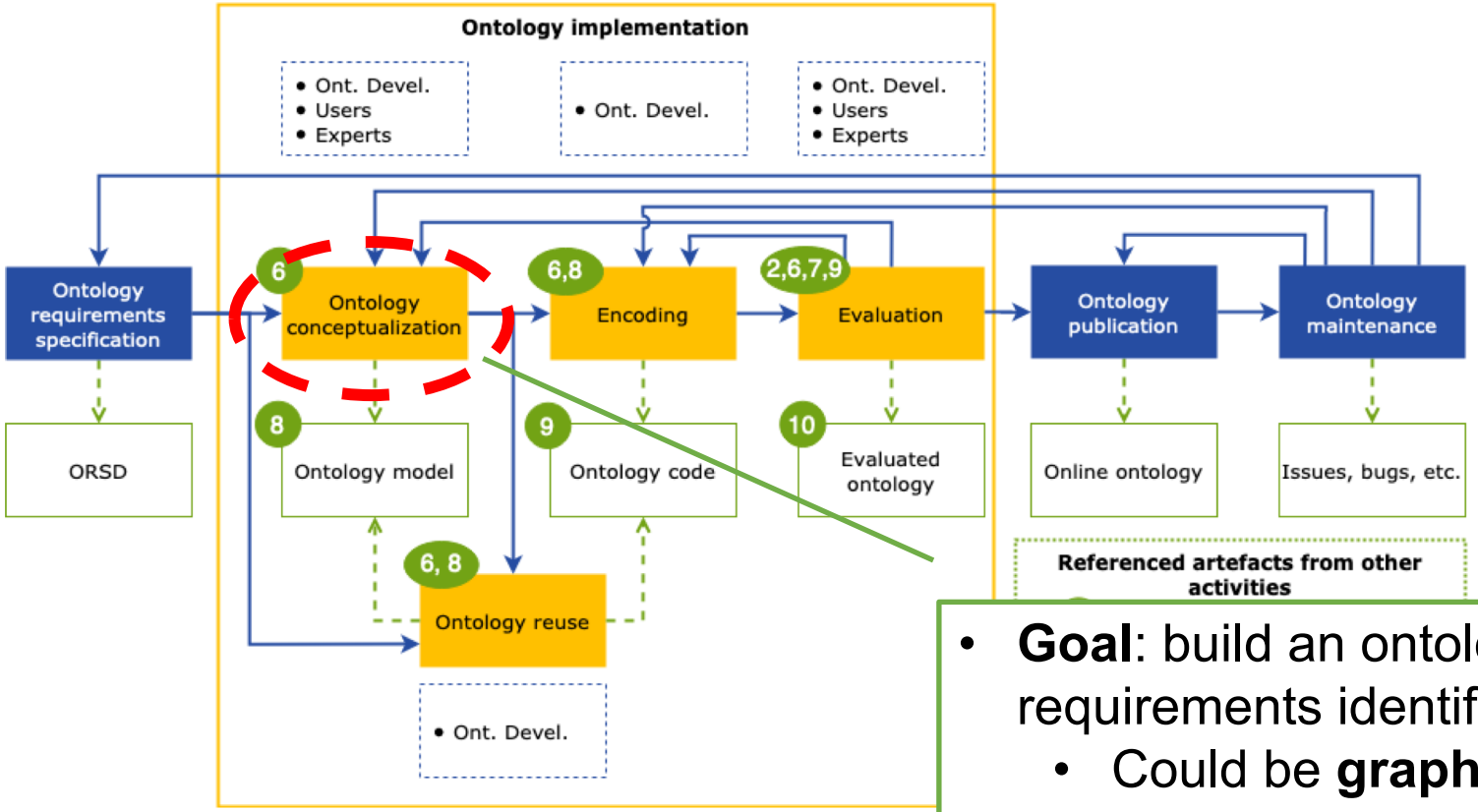
Ontology Requirements Specification Document Template	
1	**Purpose**
	The general goal of the BIMERR ontology and data model is to facilitate data sharing and interoperability among the BIMERR components through the BIMERR Interoperability Framework.
2	**Scope**
	The scope of the BIMERR ontologies is limited to the data shared through the BIF and external data sources needed in the energy efficiency domain and related domains like: KPIs, project management, weather, occupancy behavior, information objects, building geometry, building elements, materials and renovation measurements.
3	**Implementation Language**
	Ontology Web Language
4	**Intended End-Users**
	BIMERR components and application developers *BIMERR end-users and stakeholders*
5	**Intended Uses**
	Data model generation *External data sources integration*
6	**Ontology Requirements**
	a. Non-Functional Requirements
	Annotated in English *Linked to standards when possible* *Open license* *Modular* *Online availability*
	b. Functional Requirements: Groups of Competency Questions
	This section is provided for each specific domain in the confluence pages.

ORSD template
https://github.com/oeg-upm/LOT-resources/tree/master/ORSD



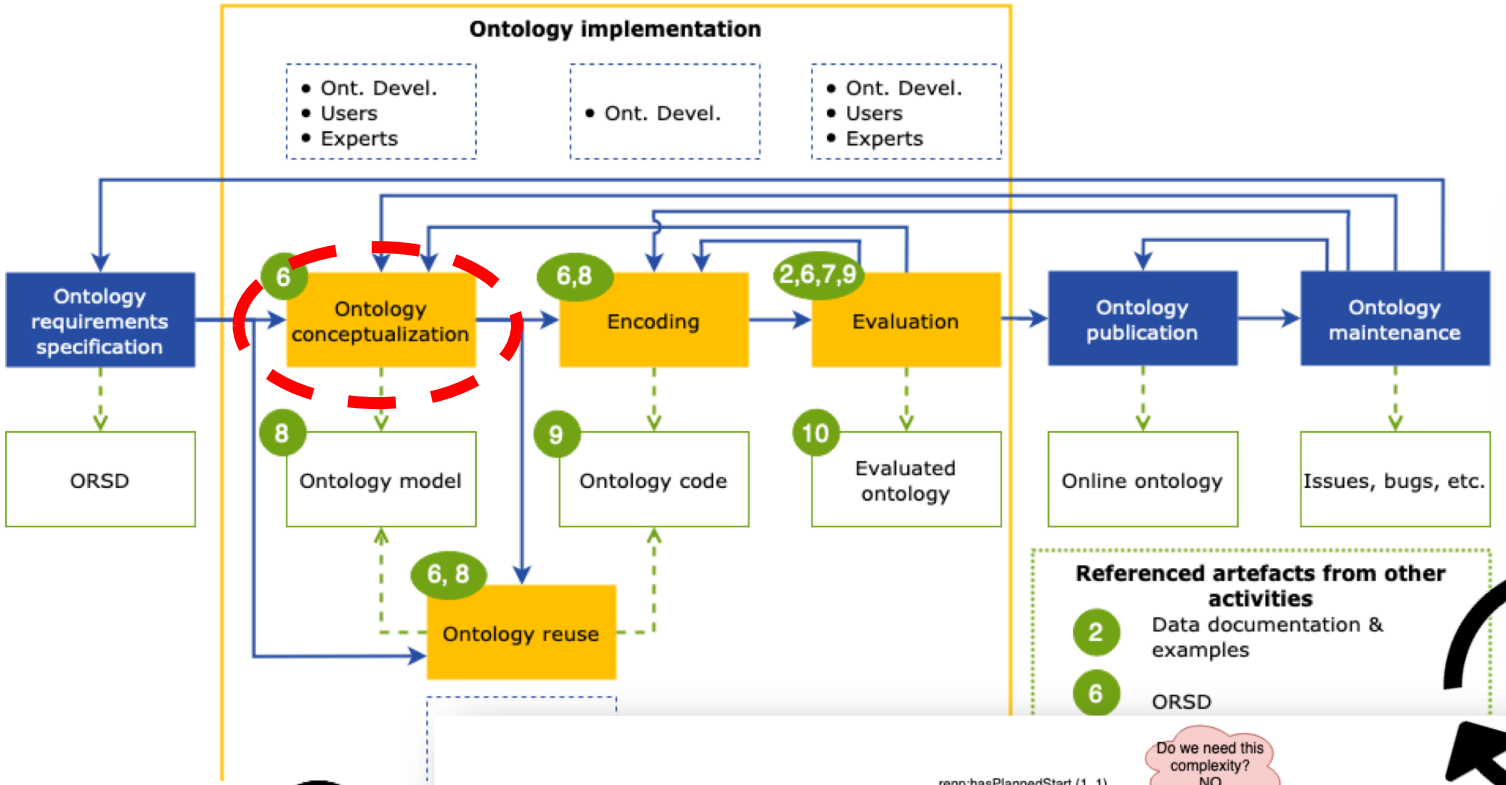


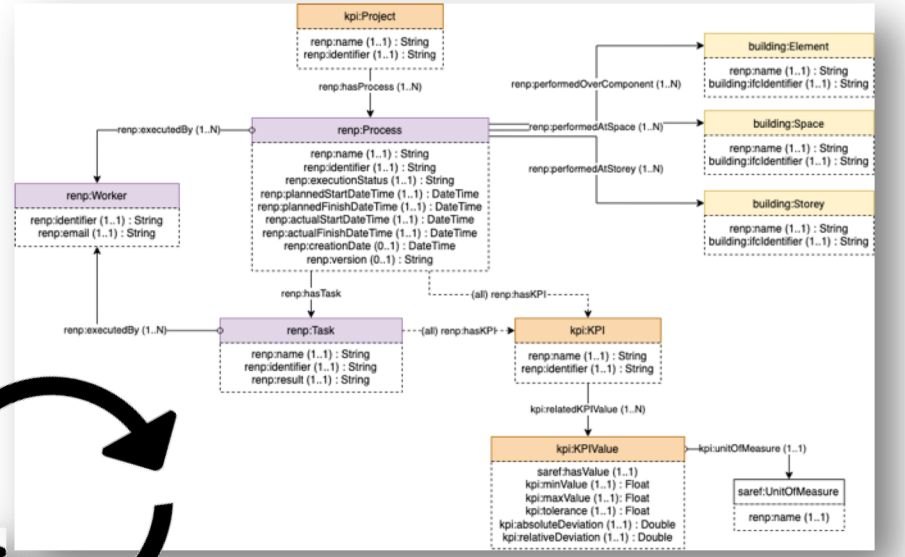

Ontology implementation
• Ont. Devel.
• Users
• Experts
• Ont. Devel.
• Ont. Devel.
• Users
• Experts
Ontology requirements specification
6
Ontology conceptualization
6,7
Ontology encoding
2,6,8
Ontology evaluation
Ontology publication
Ontology maintenance
ORSD
Ontology model
7
Ontology code
8
Evaluated ontology
9
Online ontology
Issues, bugs, etc.
6, 7
Ontology reuse
• Ont. Devel.
Referenced artefacts from other activities
2 Data documentation & examples
6 ORSD



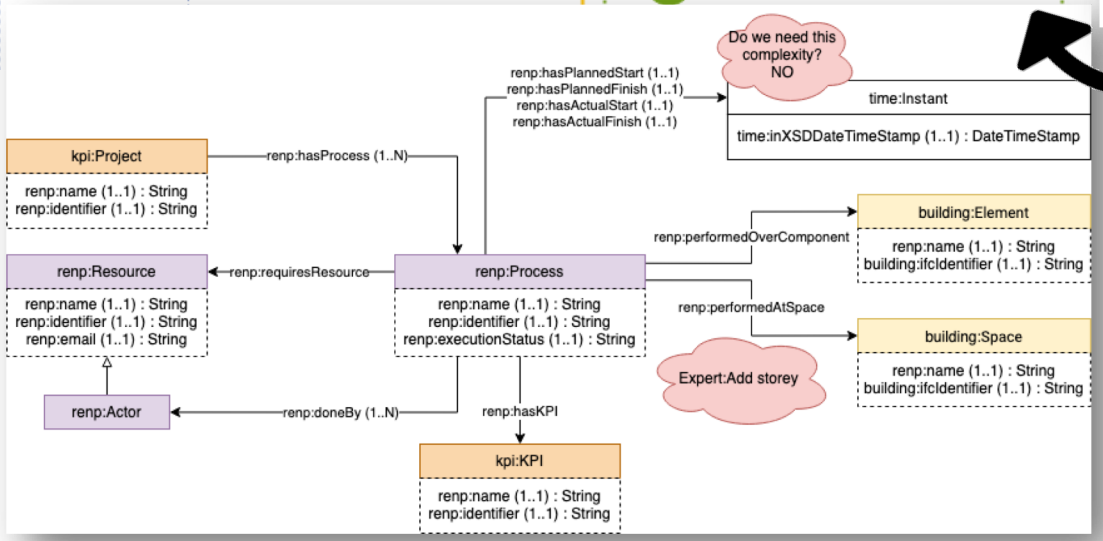
- **Goal**: build an ontology model from the ontological requirements identified.
  - Could be **graphical** or described in a formal system
- You can use
  - Blackboard
  - Pen & pencil
  - Drawing tools (diagrams.net, visio, yEd…)

Suárez-Figueroa, Mari Carmen, Asunción Gómez-Pérez, and Mariano Fernández-López. "The NeOn methodology for ontology engineering." *Ontology engineering in a networked world*. Springer, Berlin, Heidelberg, 2012. 9-34.



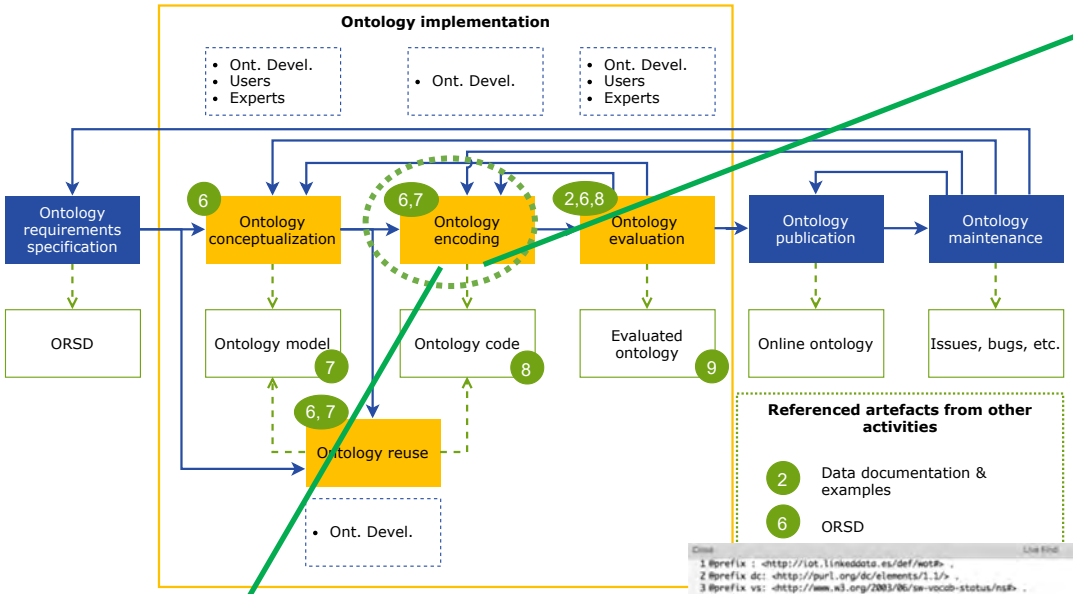


Domain Expert



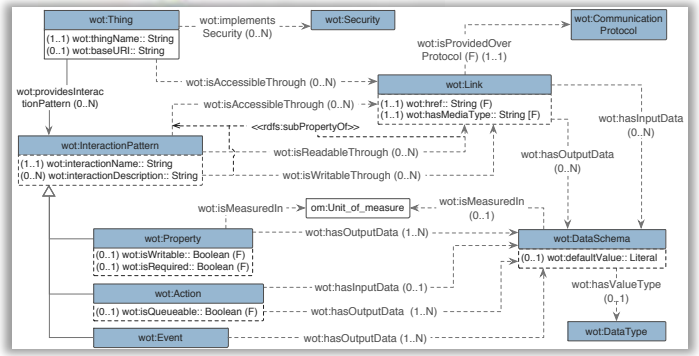
Ontology Engineer

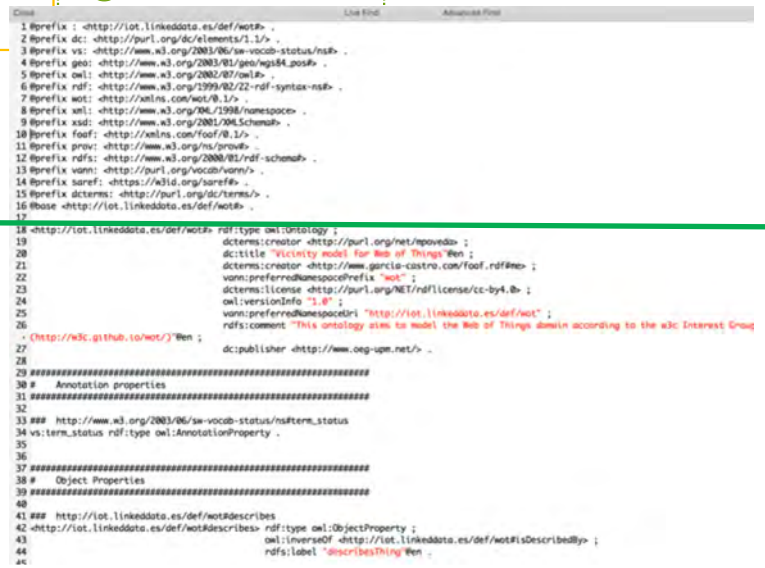
Implementation - Encoding

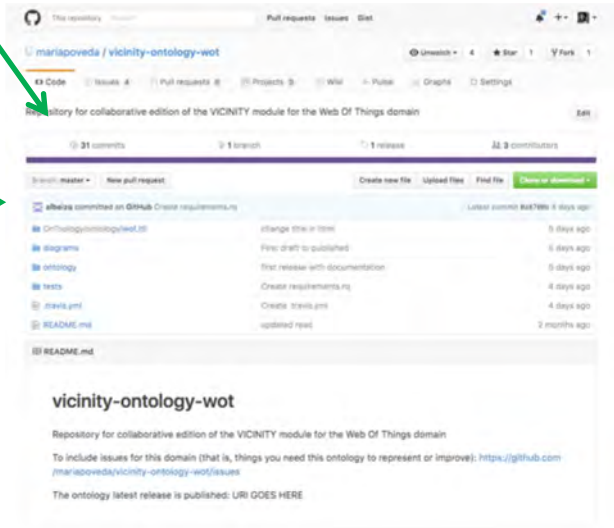


Openly managed in **GitHub**

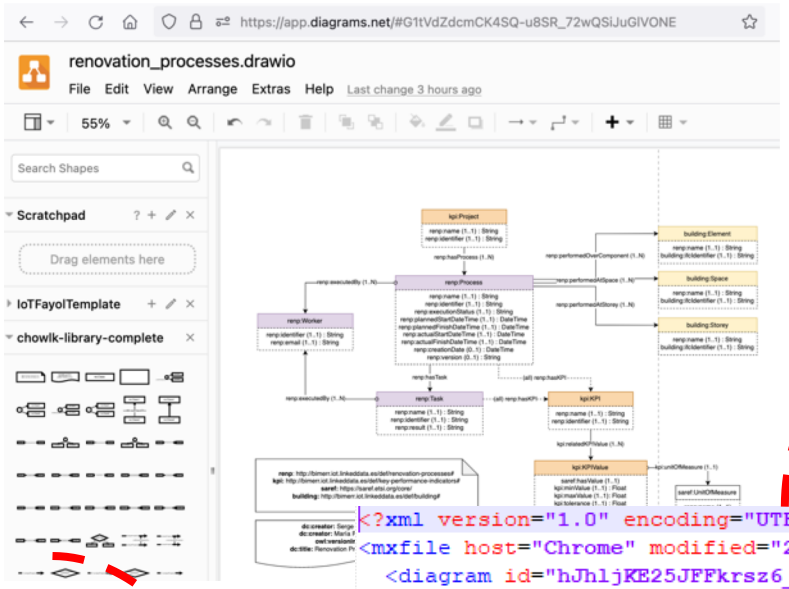
protégé







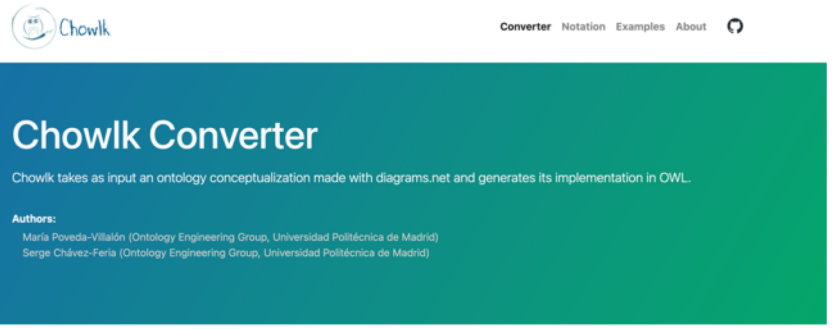
GitHub repository
https://github.com/mariapoveda/vicinity-ontology-wot

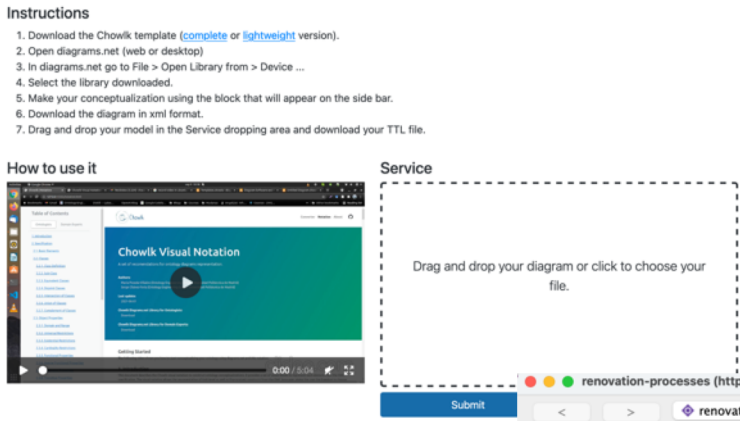


Export XML
from diagrams.net

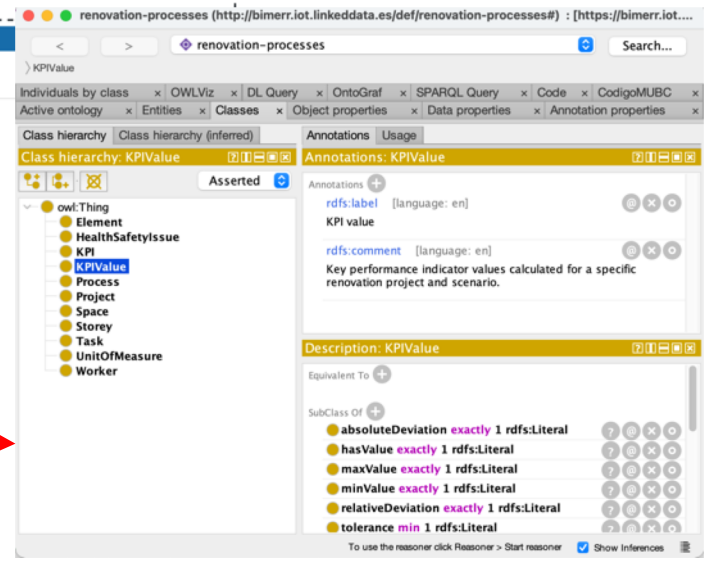
```
<?xml version="1.0" encoding="UTF-8"?>
<mxfile host="Chrome" modified="2020-11-16T14:41:32.661Z" agent="
  <diagram id="hJhljKE25JFFkrsz6_xK">
    <mxGraphModel dx="20674" dy="19061" grid="1" gridSize="1" gui
      <root>
        <mxCell id="0" />
        <mxCell id="1" parent="0" />
        <mxCell id="2" style="edgeStyle=orthogonalEdgeStyle;rounc
          <mxGeometry relative="1" as="geometry">
            <mxPoint x="-17271" y="-17490" as="targetPoint" />
          </mxGeometry>
        </mxCell>
        <mxCell id="3" value="building:Building" style="rounded=(
          <mxGeometry x="-17377" y="-17470.239999999998" width="2
        </mxCell>
        <mxCell id="4" value="(all) bot:hasStorey" style="edgeSty
          <mxGeometry x="-0.1468" y="2" relative="1" as="geometry
            <Array as="points">
              <mxPoint x="-17470" y="-17457" />
              <mxPoint x="-17470" y="-17403" />
            </Array>
            <mxPoint as="offset" />
          </mxGeometry>
        </mxCell>
```

Convert into OWL
with Chowlk





Continue in Protégé
(if you want)

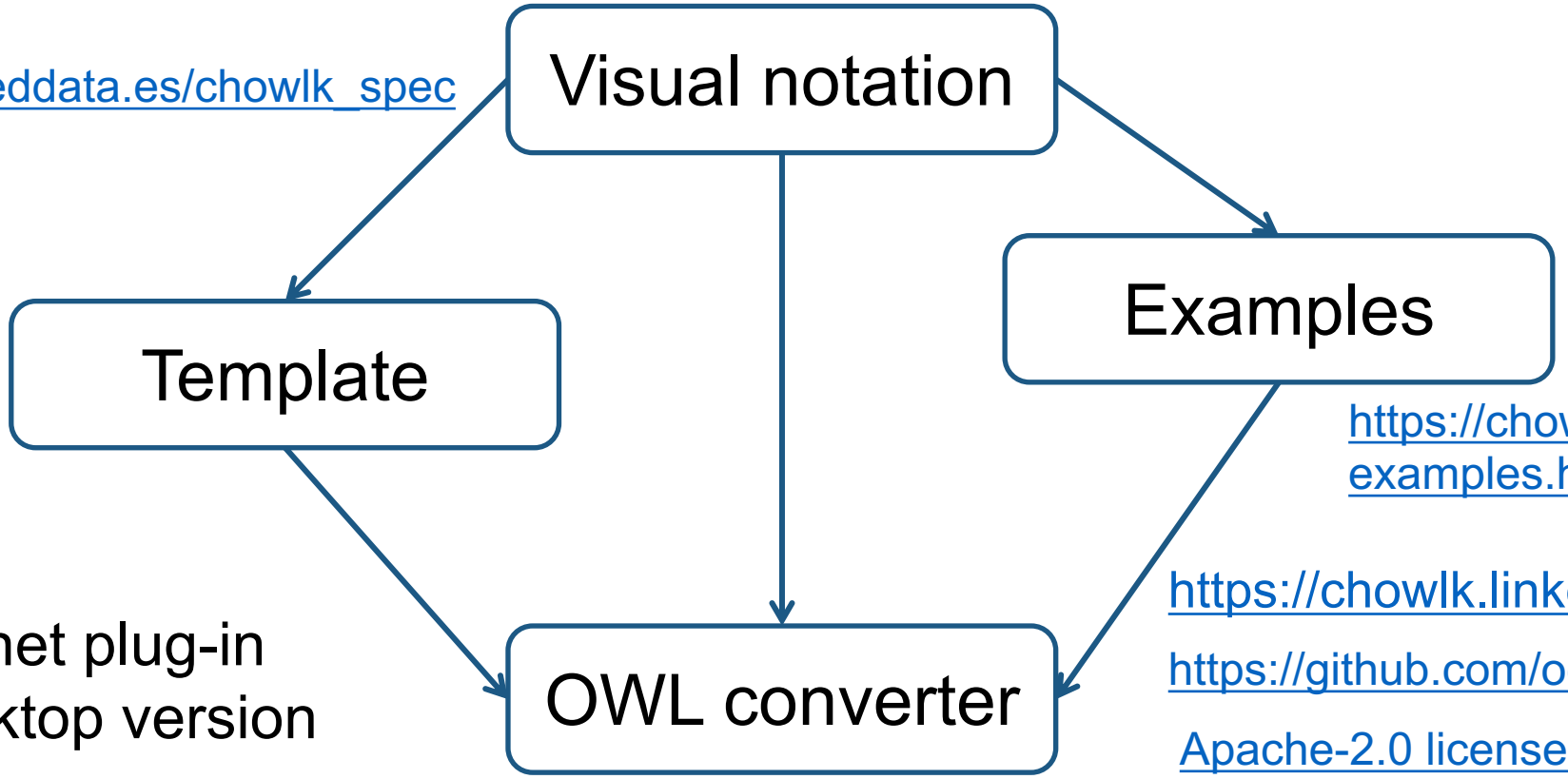


Chowlk

BIMERR RENOVATION 4.0

COGITO

https://chowlk.linkeddata.es/chowlk_spec



https://chowlk.linkeddata.es/examples.html

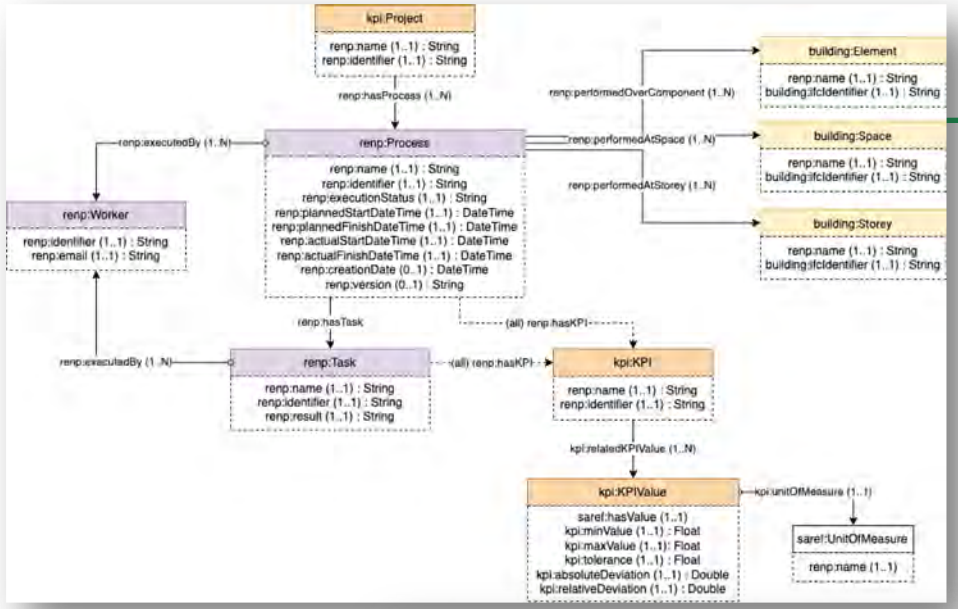
+ diagrams.net plug-in
Only for desktop version

https://chowlk.linkeddata.es

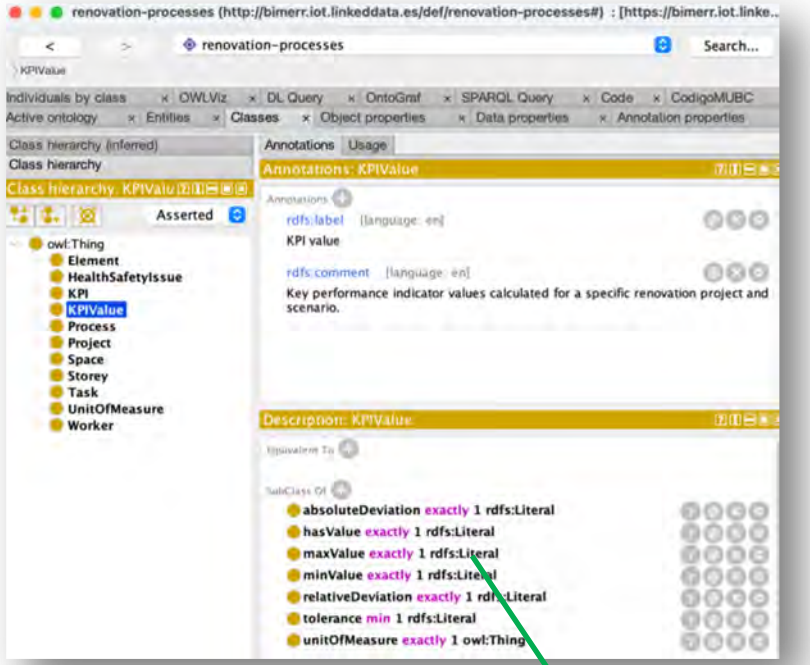
https://github.com/oeg-upm/Chowlk

Apache-2.0 license

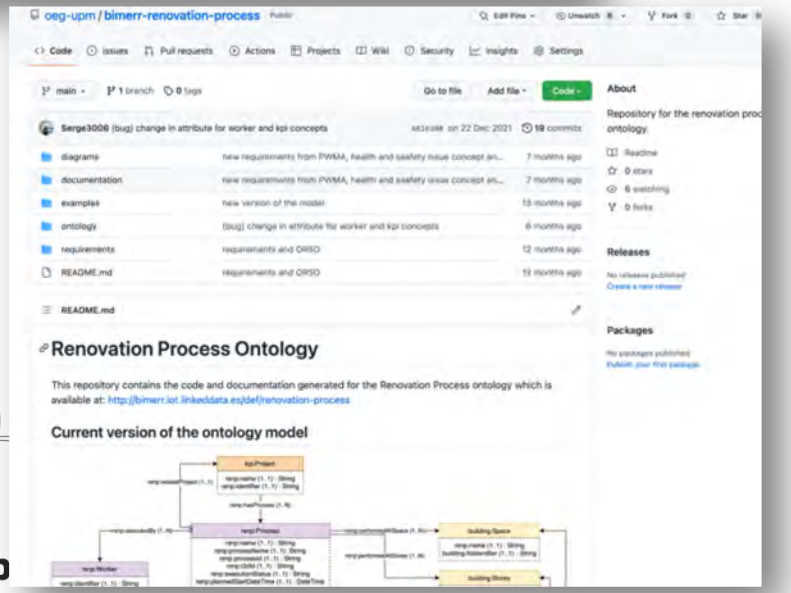
Implementation - Encoding



Chowlk



protégé



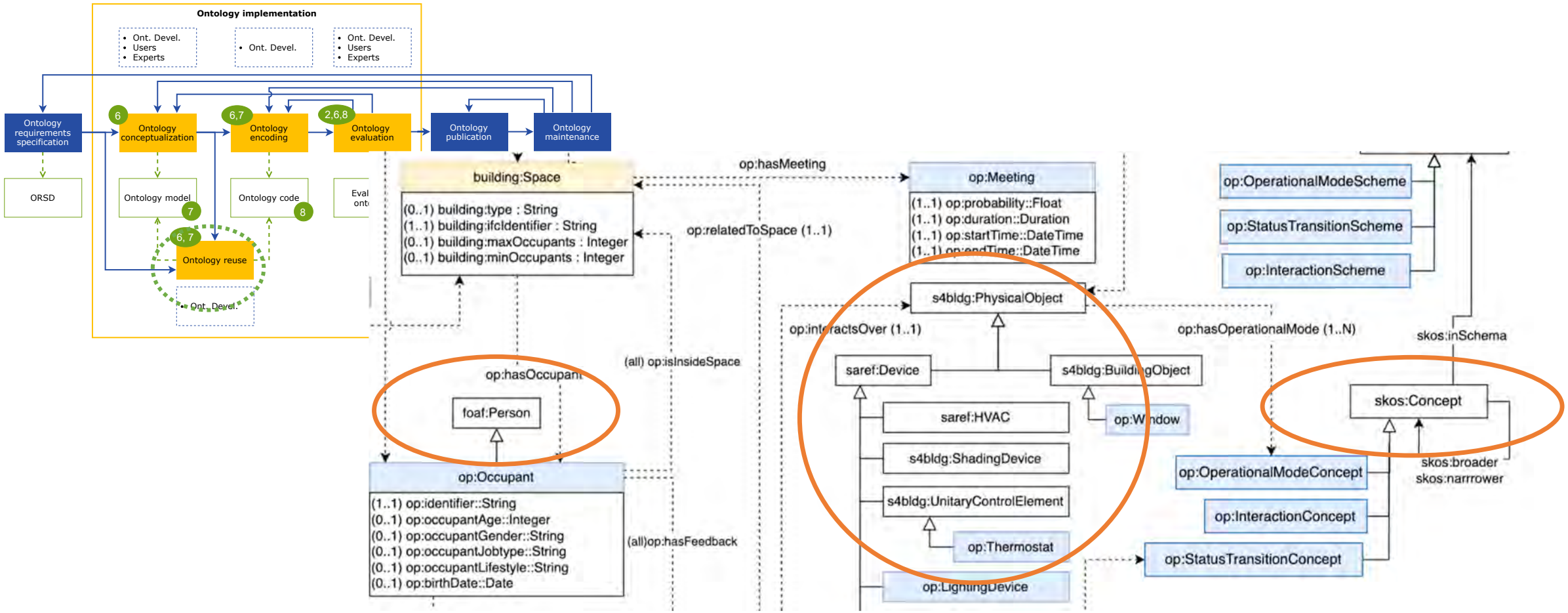
GitHub

```
@prefix : <http://bimerr.iot.linkeddata.es/def/renovation-processes#> .
@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#> .
@base <http://bimerr.iot.linkeddata.es/def/renovation-processes#> .

<http://bimerr.iot.linkeddata.es/def/renovation-processes#> rdf:type owl:Ontology ;
                                                             <http://purl.org/dc/elements/1.1/creator>
                                                             "María Poveda-Villalón" ,
                                                                      "Serge Chávez-Feria" ;
                                                             <http://purl.org/dc/elements/1.
                                                             1/description> "Ontology code created by
                                                             Chowlk" ;
                                                             <http://purl.org/dc/elements/1.1/license>
                                                             http://purl.org/NET/rdflicense/cc-by4.0" ;
                                                             <http://purl.org/dc/elements/1.1/publisher
                                                              "http://www.oeg-upm.net/" ;
                                                             <http://purl.org/dc/elements/1.1/title>
                                                             "Renovation Processes and Work Orders
                                                             Ontology" ;
                                                             <http://purl.
                                                             org/vocab/vann/preferredNamespacePrefix>
                                                             "renp" ;
                                                             rdfs:comment "Ontology to model building
                                                             renovation processes and work orders sent
                                                             to the workers on-site."@en ;
                                                             owl:versionInfo "0.0.3" .

#################################################################
#    Annotation properties
#################################################################

###  http://purl.org/dc/elements/1.1/creator
```



- **Reusing knowledge resources**

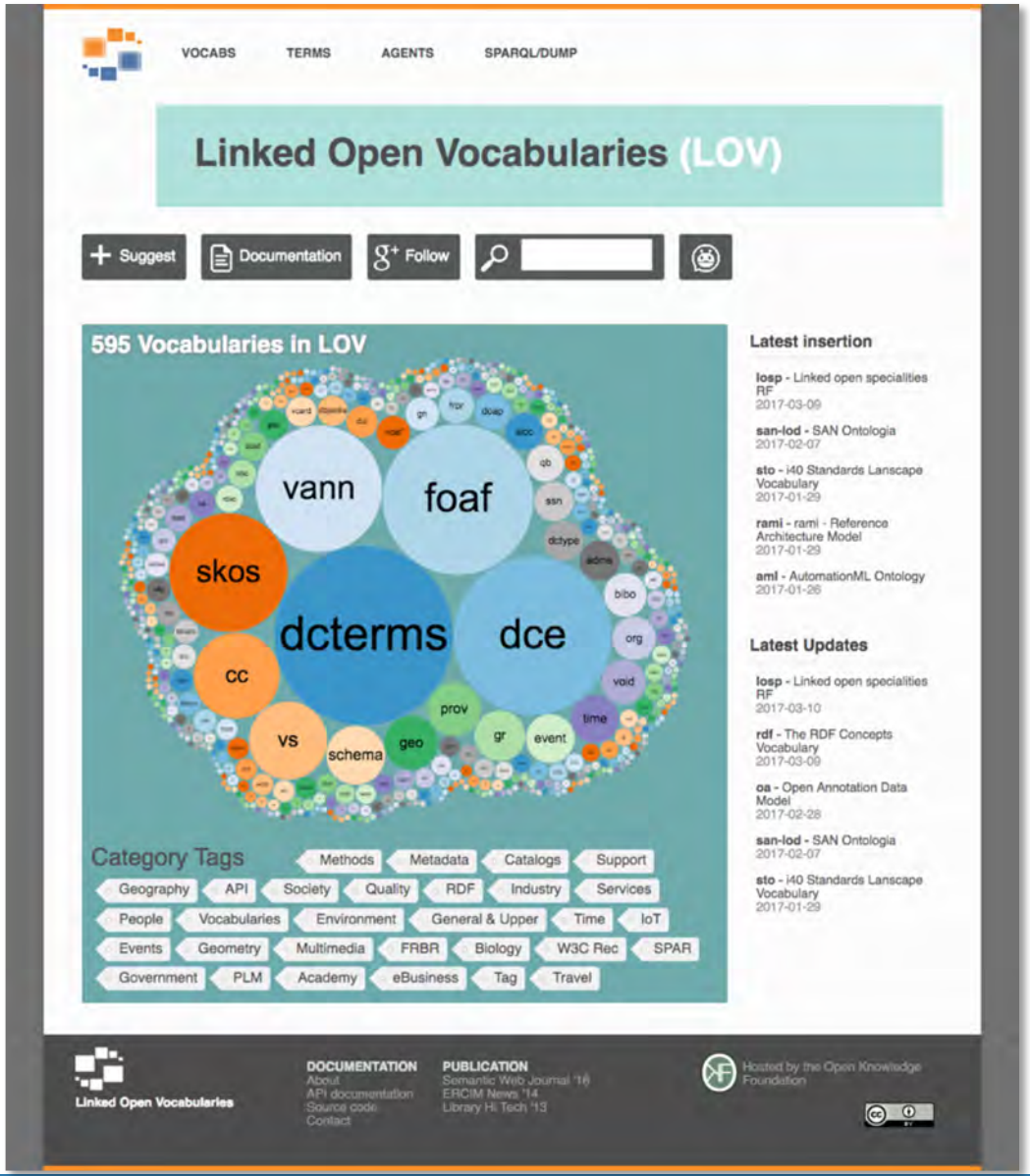
XML … Google refine

Look for existing ontologies:
https://lov.linkeddata.es
Etc.

https://lov.linkeddata.es

- **Mission: promote and facilitate the reuse of well documented vocabularies in the Linked Data ecosystem**

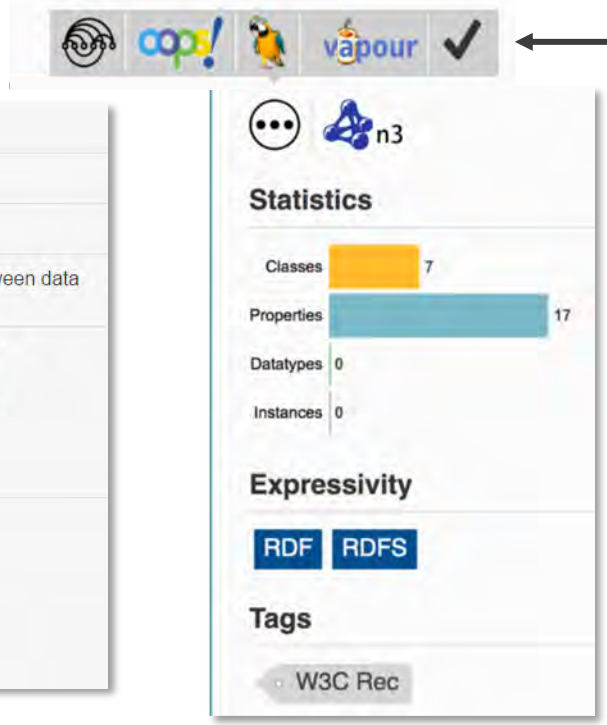
- Vocabularies registry and index
- Datalift
  - http://datalift.org/
- Started at 2011
- Hosted by OEG



Data Catalog Vocabulary (dcat)

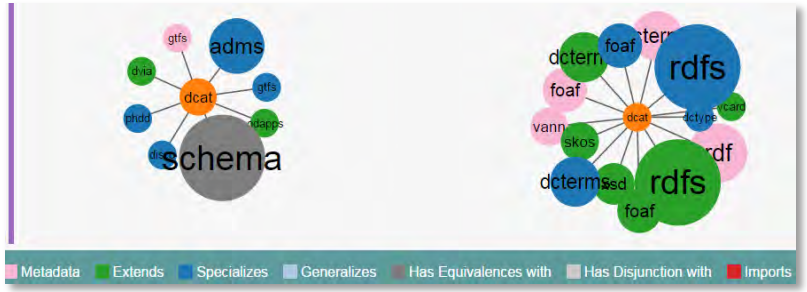
Metadata

URI	http://www.w3.org/ns/dcat
Namespace	http://www.w3.org/ns/dcat#
homepage	http://www.w3.org/TR/vocab-dcat/
Description	DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web @en
Language	Arabic ar, Greek el, English en, Spanish es, French fr, Japanese ja
Contributor	Richard Cyganiak http://google.com/+RichardCyganiak; Phil Archer https://plus.google.com/103670676337547906055; Fadi Maali

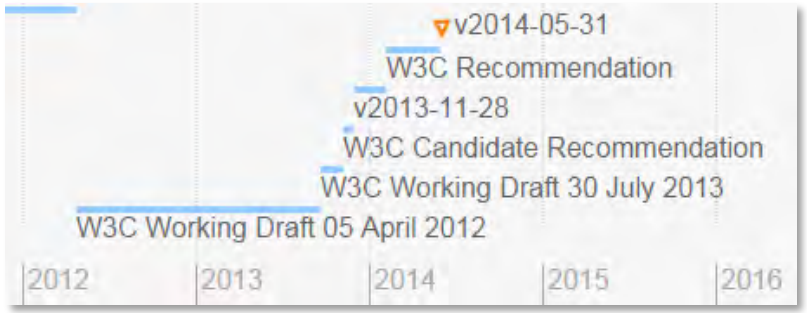


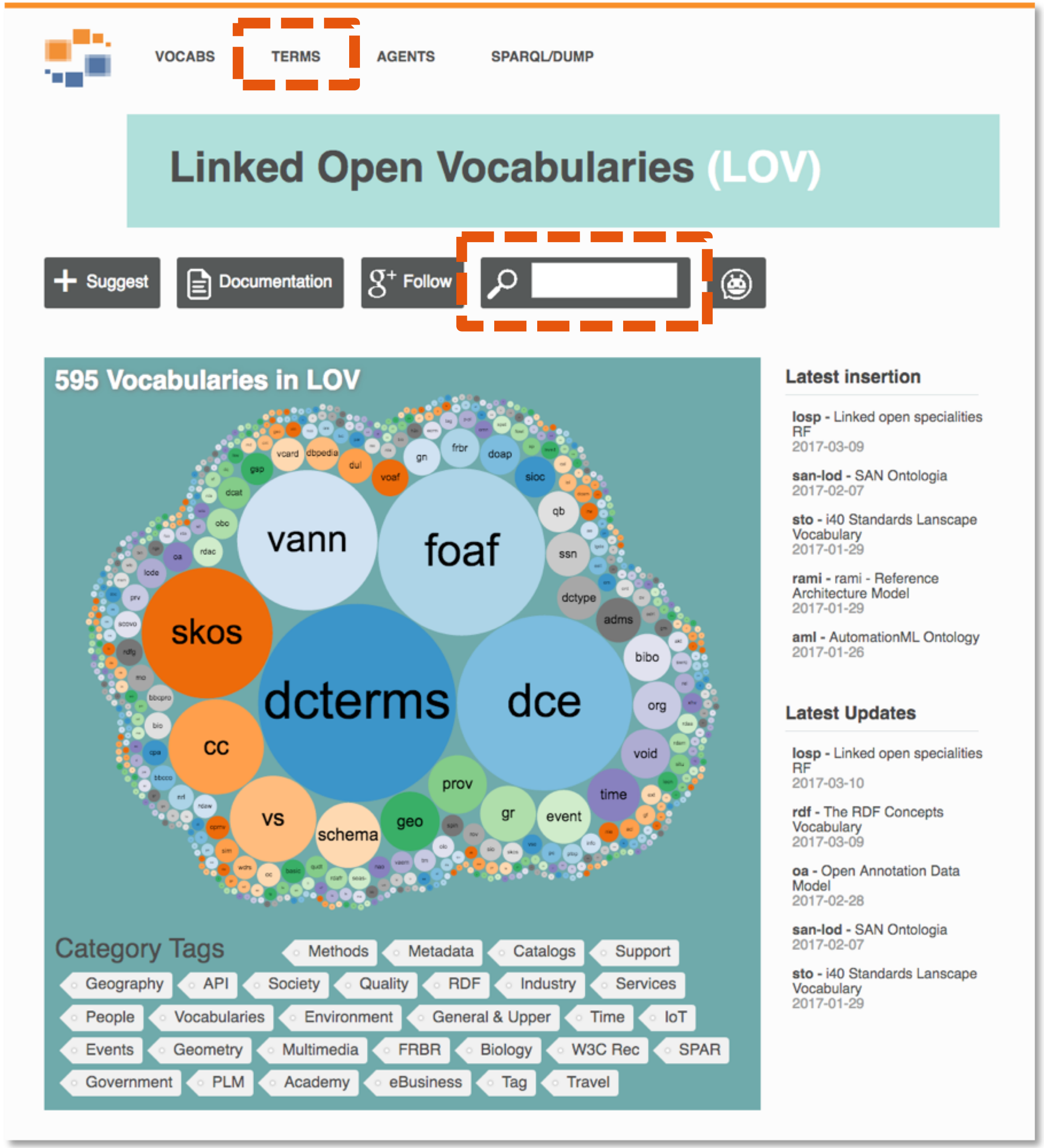
Connection with other applications

Relationships

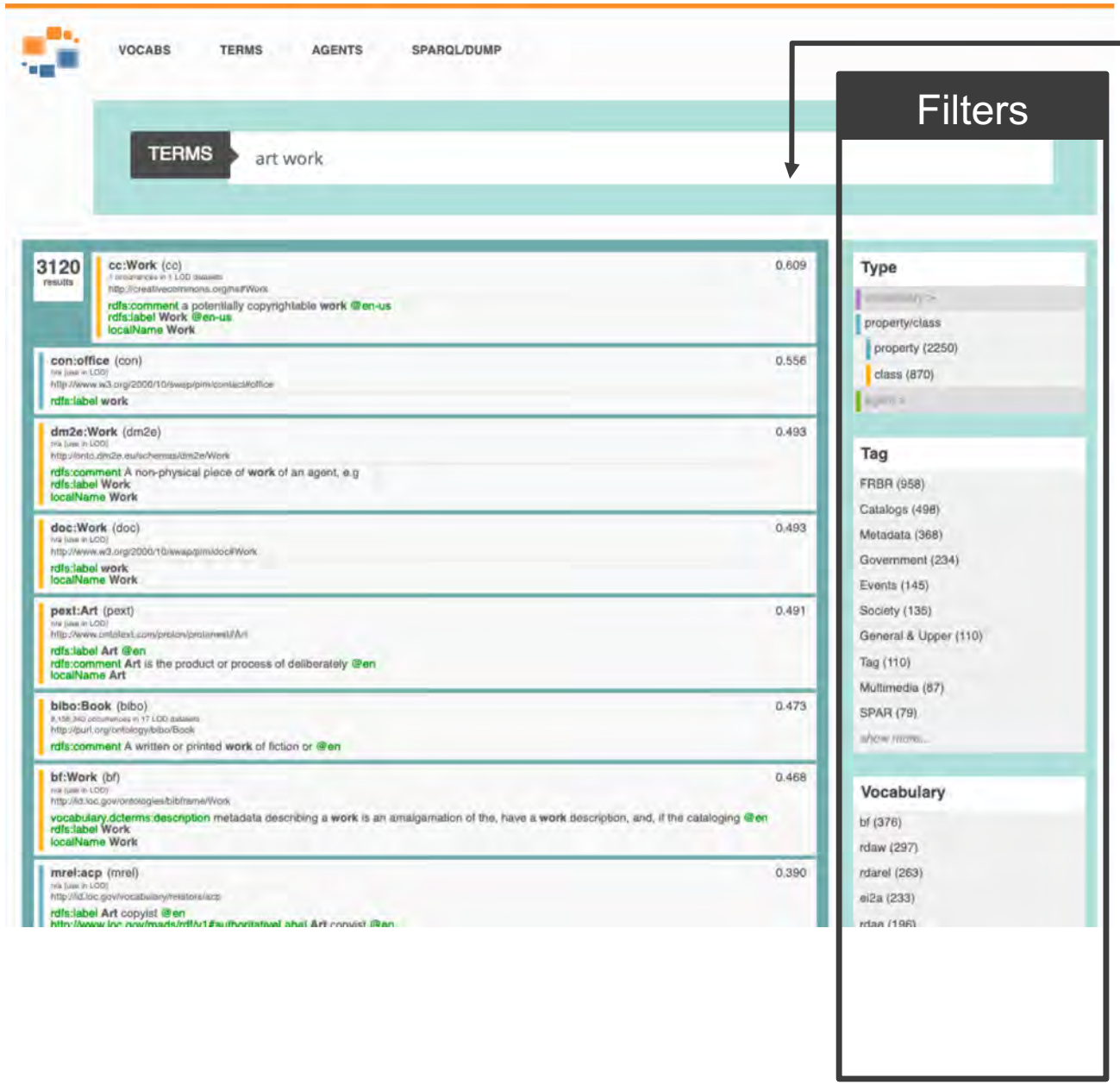


Versions



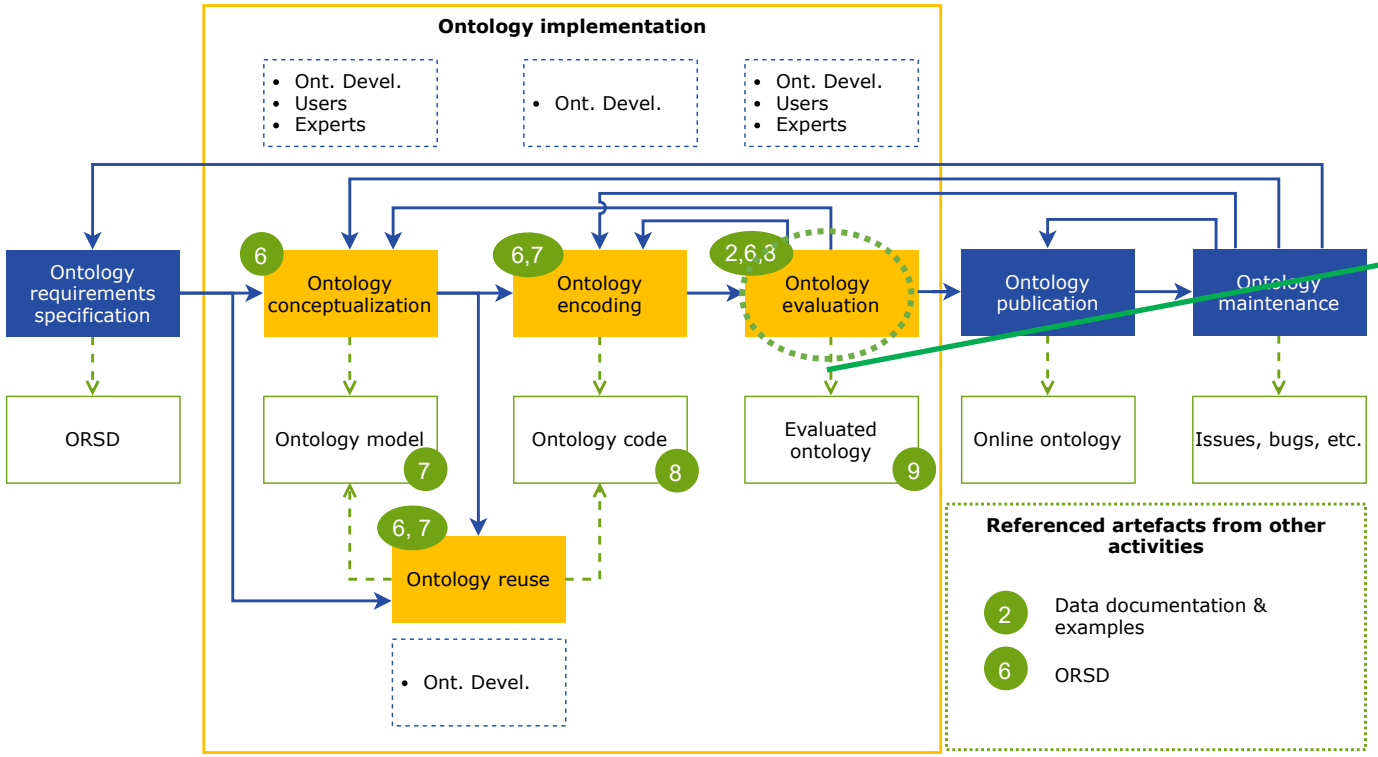

VOCABS
TERMS
AGENTS
SPARQL/DUMP
Linked Open Vocabularies (LOV)
Suggest
Documentation
Follow
595 Vocabularies in LOV
vann
foaf
skos
dcterms
dce
cc
vs
schema
geo
prov
gr
event
time
void
org
bibo
adms
dctype
ssn
qb
sioc
doap
frbr
gn
voaf
dul
vcard
dbpedia
Category Tags
Methods
Metadata
Catalogs
Support
Geography
API
Society
Quality
RDF
Industry
Services
People
Vocabularies
Environment
General & Upper
Time
IoT
Events
Geometry
Multimedia
FRBR
Biology
W3C Rec
SPAR
Government
PLM
Academy
eBusiness
Tag
Travel
Latest insertion
losp - Linked open specialities RF
2017-03-09
san-lod - SAN Ontologia
2017-02-07
sto - i40 Standards Lanscape Vocabulary
2017-01-29
rami - rami - Reference Architecture Model
2017-01-29
aml - AutomationML Ontology
2017-01-26
Latest Updates
losp - Linked open specialities RF
2017-03-10
rdf - The RDF Concepts Vocabulary
2017-03-09
oa - Open Annotation Data Model
2017-02-28
san-lod - SAN Ontologia
2017-02-07
sto - i40 Standards Lanscape Vocabulary
2017-01-29

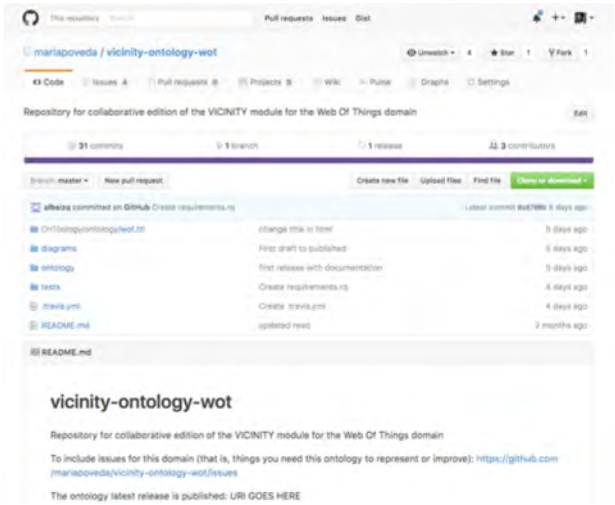
Search terms



Ranked

- Term appearing in primary and secondary annotations
- Vocabulary popularity in LOV
- Term use in LOD



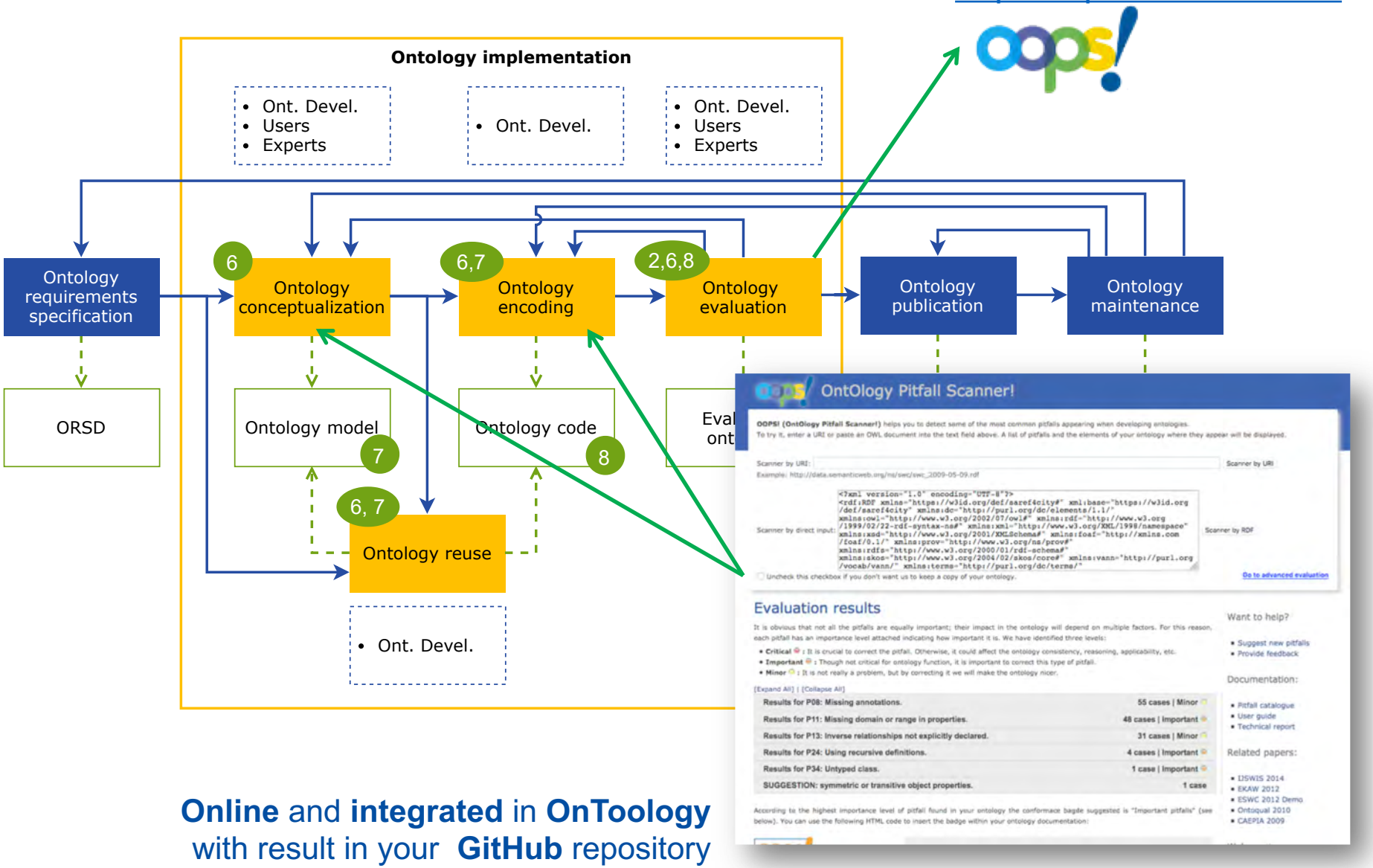


Online and **notifications** in **GitHub** repository
https://github.com/mariapoveda/vicinity-ontology-wot

- It refers to the activity of checking the technical quality of an ontology against a frame of reference. [NeOn]
  - Logical consistency checking
  - Domain coverage
  - Check common errors → **OOPS!** (http://oops.linkeddata.es/)
  - Check functional requirements → **Themis** (http://themis.linkeddata.es)

Implementation - Evaluation

http://oops.linkeddata.es

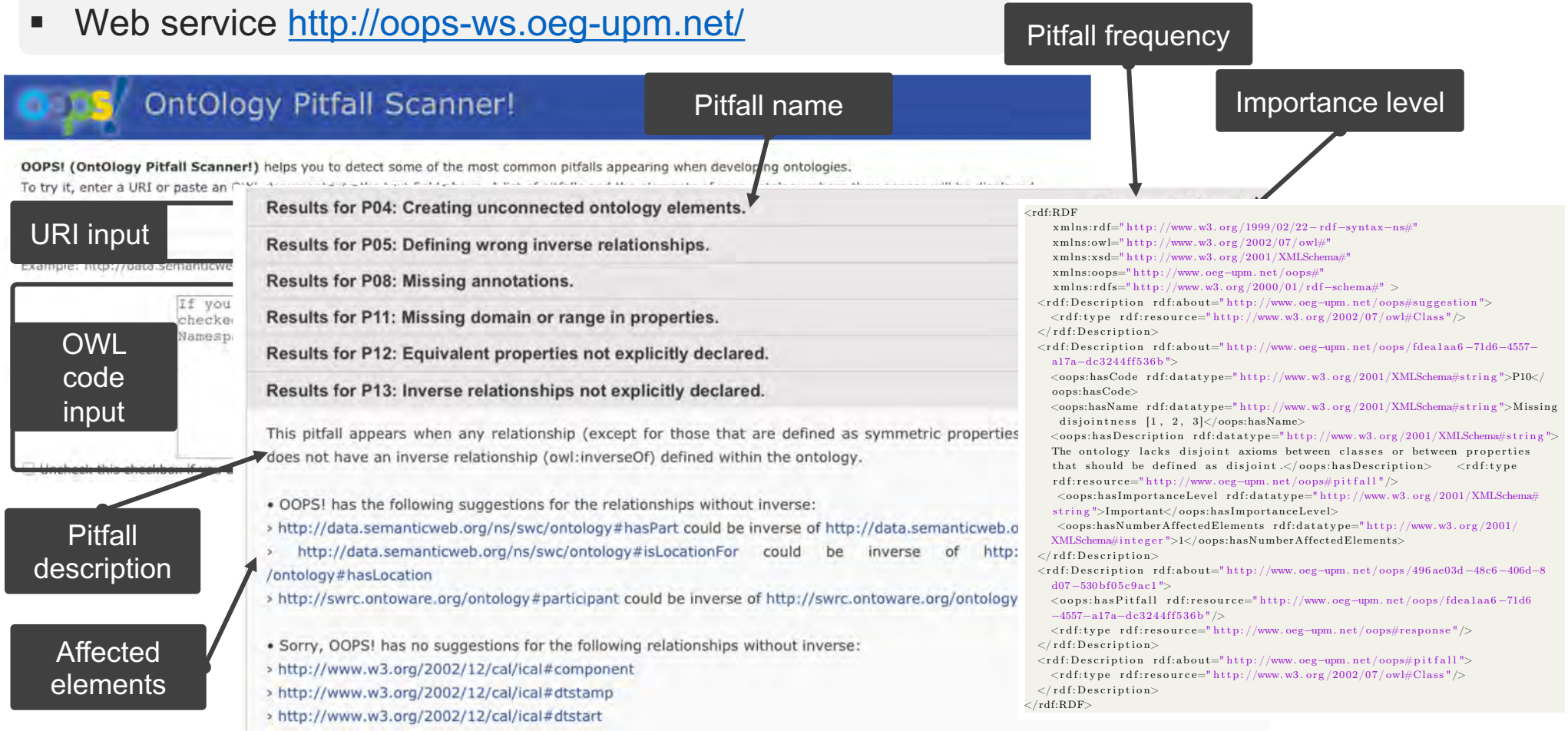


Online and **integrated** in **OnToology** with result in your **GitHub** repository

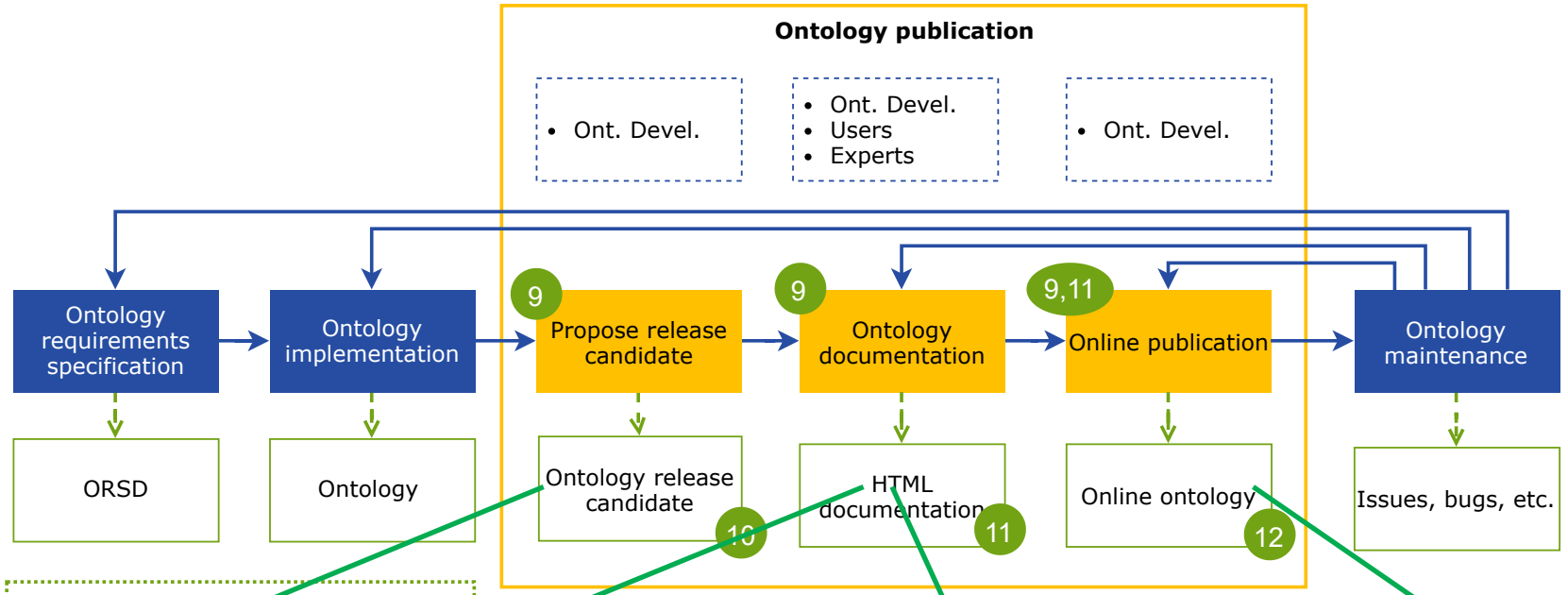
Slide taken from “Trendy Practices and Tools in Ontological Engineering” by Dr. María Poveda Villalón

Evaluation - OOPS! – OntOlogy Pitfall Scanner!

- Implements the **48** detection methods for **33 pitfalls**
  - Pitfalls selection
  - Selection by dimensions and aspects
- Web user interface http://oops.linkeddata.es/
- Web service http://oops-ws.oeg-upm.net/

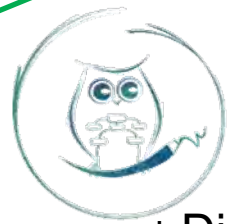


Slide taken from “Trendy Practices and Tools in Ontological Engineering” by Dr. María Poveda Villalón



GitHub

- HTML generation from OWL code
- Multilingual
- Separated sections

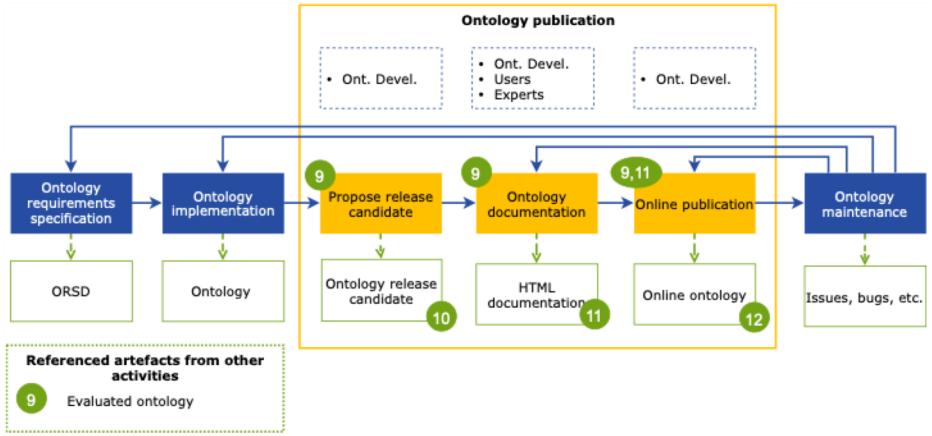


Chowlk

+ Diagrams and examples (Some ideas: https://arxiv.org/abs/2003.13084)
+ Some tools:
  + Draw.io
  + yEd graph editor
  + Microsoft Visio

- Own URI
- purl, w3id, etc.
- Content negotiation
- Registry

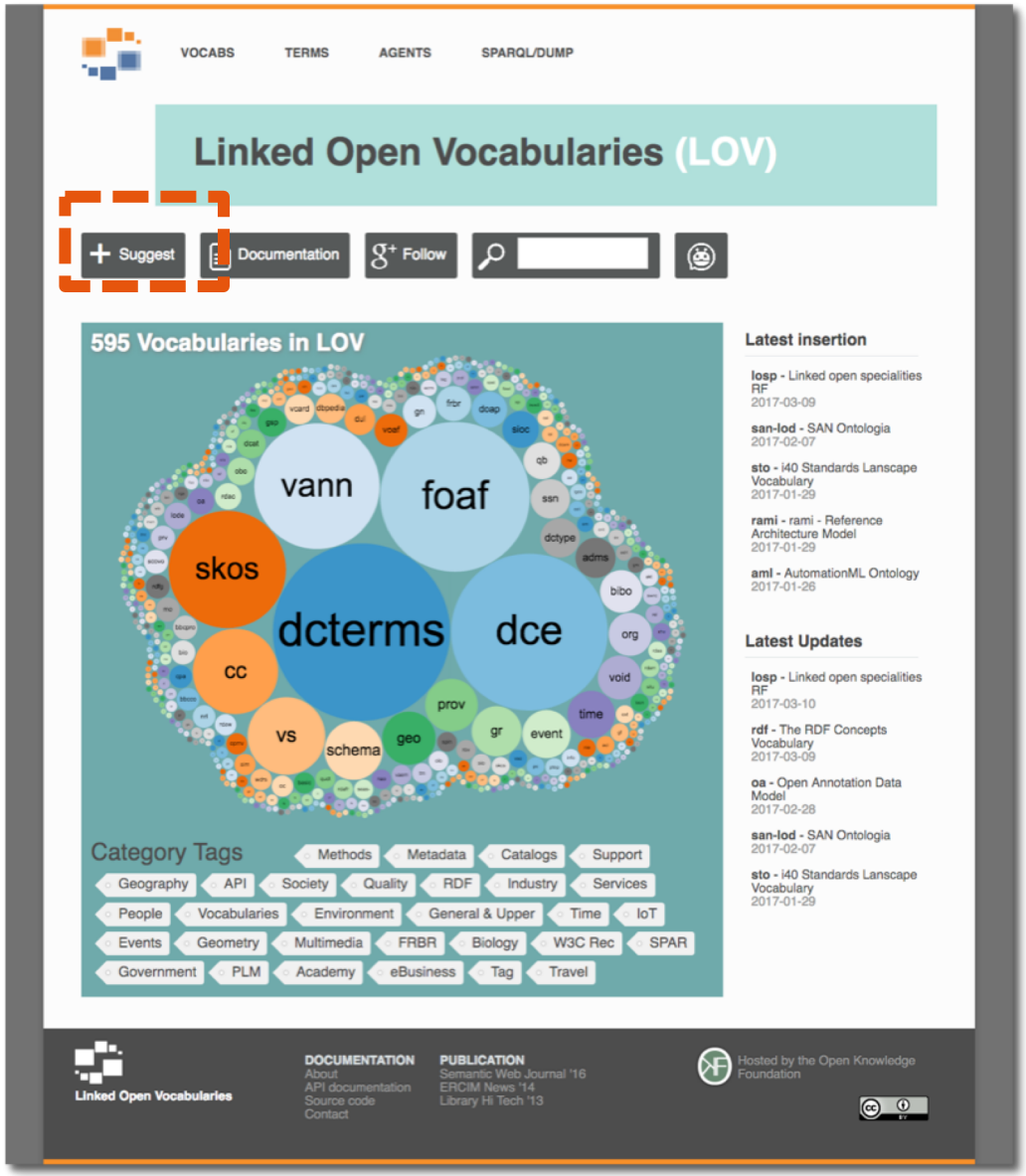
Publication – Share your ontology



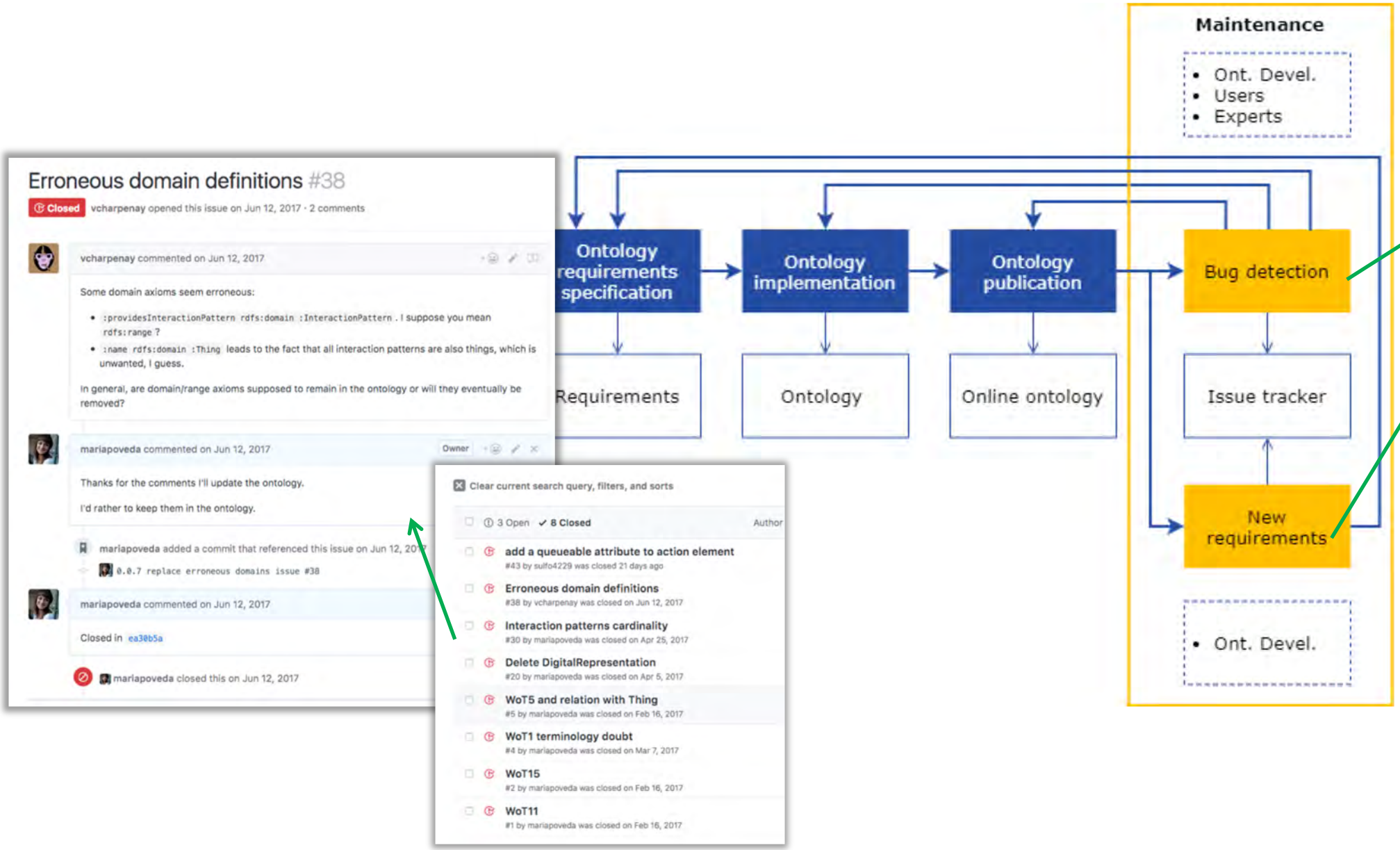
- **Mission: promote and facilitate the reuse of well documented vocabularies in the Linked Data ecosystem**

- Vocabularies registry and index
- Datalift
  - http://datalift.org/
- Started at 2011
- Hosted by OEG

https://lov.linkeddata.es



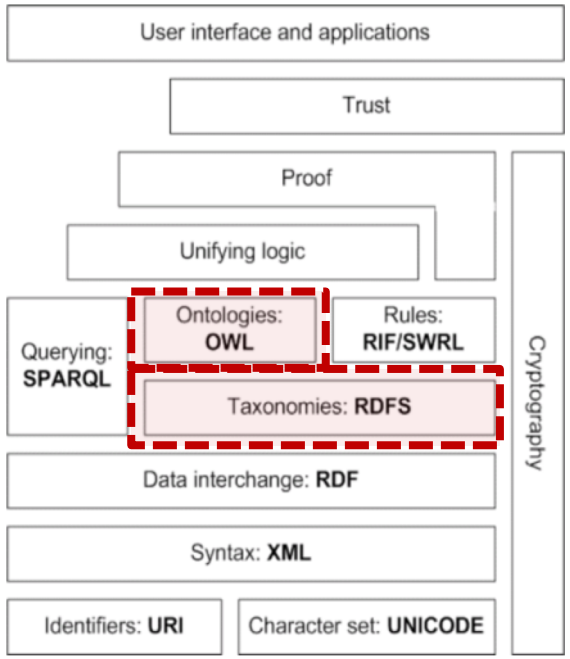
Maintenance

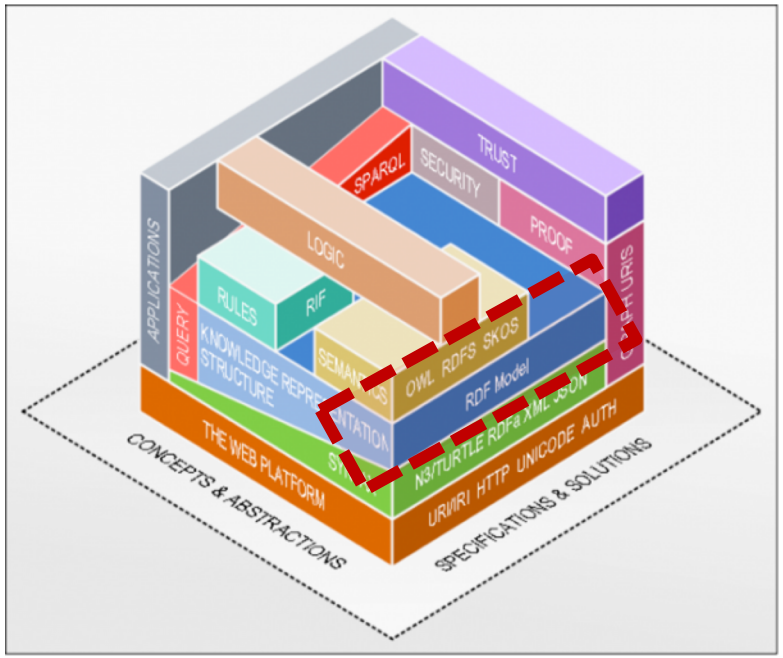


Openly reported in **GitHub issue tracker**: new needs, bugs, etc.

https://cogito.iot.linkeddata.es/

This slide has been taken from Raúl García Castro presentation at EMSE


User interface and applications
Trust
Proof
Unifying logic
Querying: SPARQL
Ontologies: OWL
Rules: RIF/SWRL
Taxonomies: RDFS
Data interchange: RDF
Syntax: XML
Identifiers: URI
Character set: UNICODE
Cryptography


APPLICATIONS
QUERY
RULES
RIF
KNOWLEDGE REPRESENTATION STRUCTURE
SEMANTICS
OWL RDFS SKOS
RDF Model
N3/TURTLE RDFa XML JSON
URI/IRI HTTP UNICODE AUTH
THE WEB PLATFORM
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS
SPARQL
SECURITY
TRUST
PROOF
LOGIC
SYNTAX

- RDF: **R**esource **D**escription **F**ramework

	Database	XML	RDF(S)	
Schema				RDF Schema
Data				RDF

- W3C Recommendation
  - Model
  - Syntax
  - Semantics

Slide taken from “RDF, RDF Schema y SPARQL” by O. Corcho, R. García-Castro”

Important message

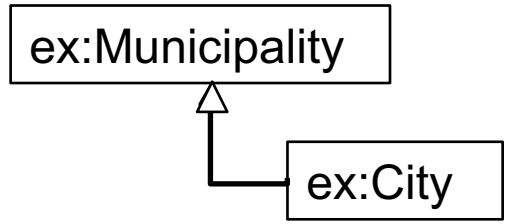
- You **can't coin** names in someone else's **namespace**
- RDF, RDF(S) and OWL provide standard constructs
  - → you **don't** need to **re-invent** them, you just provide the domain terms
    - And if you do, no one else would understand ☺

Classes

- **rdfs:Class**
  - Concepts of the domain (generally)
  - **Classes with name**:
    - URI as identifier

RDF declaration	Diagram proposal
ex:City —rdf:type→ rdfs:Class	ex:City

- **rdfs:subClassOf**
  - The individuals belonging to a class also belong to the parent classes in the hierarchy

RDF declaration	Diagram proposal
ex:City —rdfs:subClassOf→ ex:Municipality	

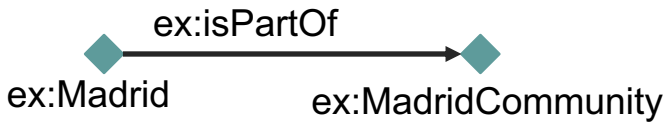
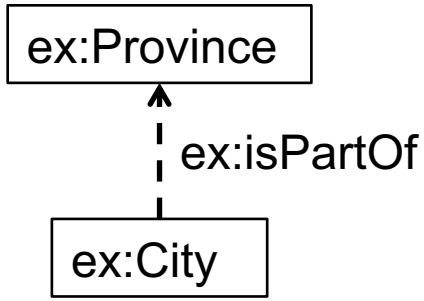
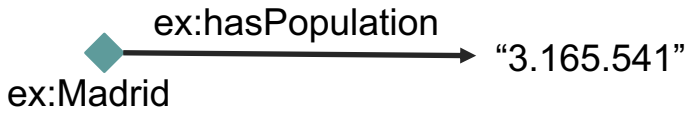
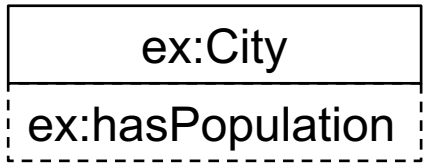
1. Define the classes in the proposed domain

2. Define hierarchies in the proposed domain

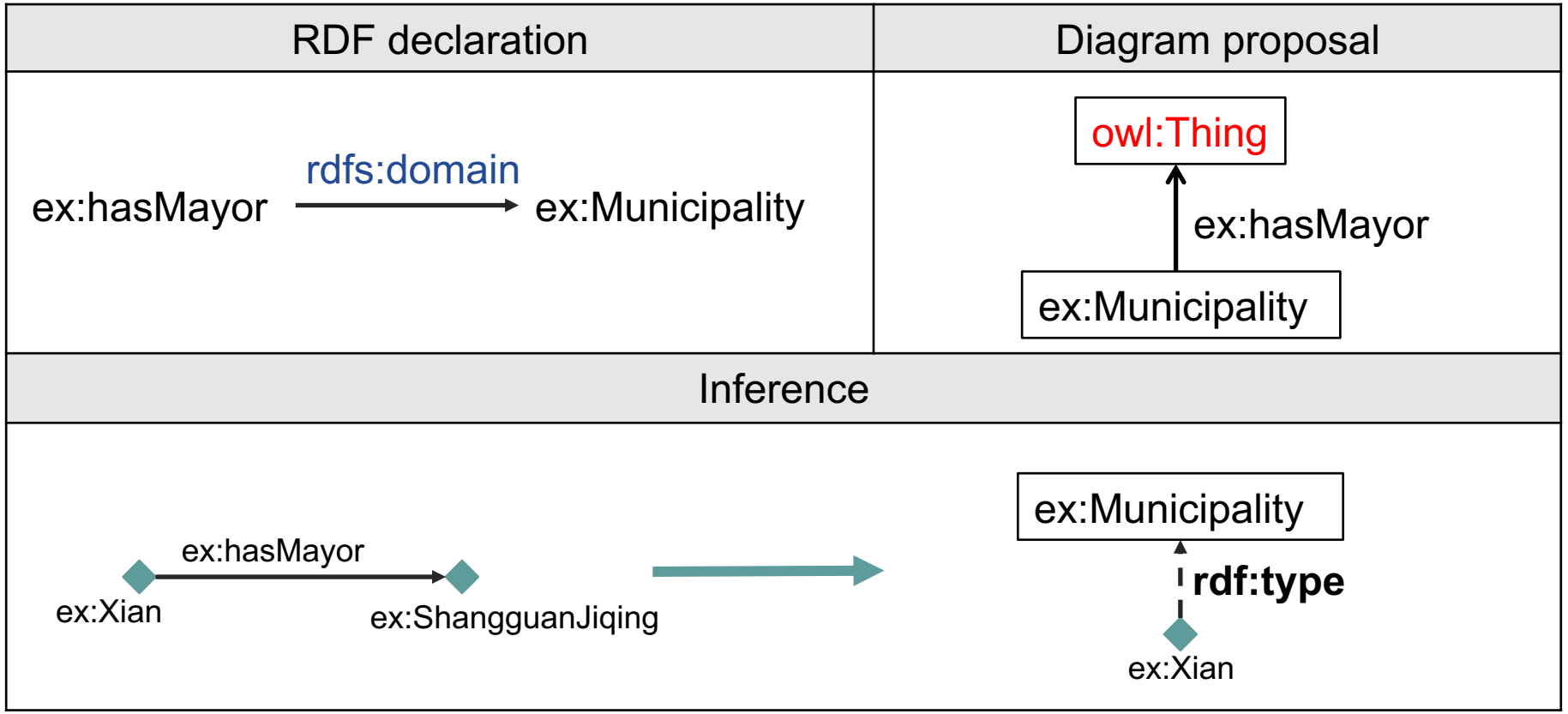
- Universities offer subjects
- A subject is delivered in one or more groups
- A subject for a given group is taught by only 1 professor
  - But in some departments this restriction does not apply
- A subject is offered only for 1 degree
- A university located at some address
- An address has a street, a number and a postal code
- An address is located in a municipality
- A municipality could have borders with other municipalities
- A professor could be associate or assistant but not at the same time

Properties

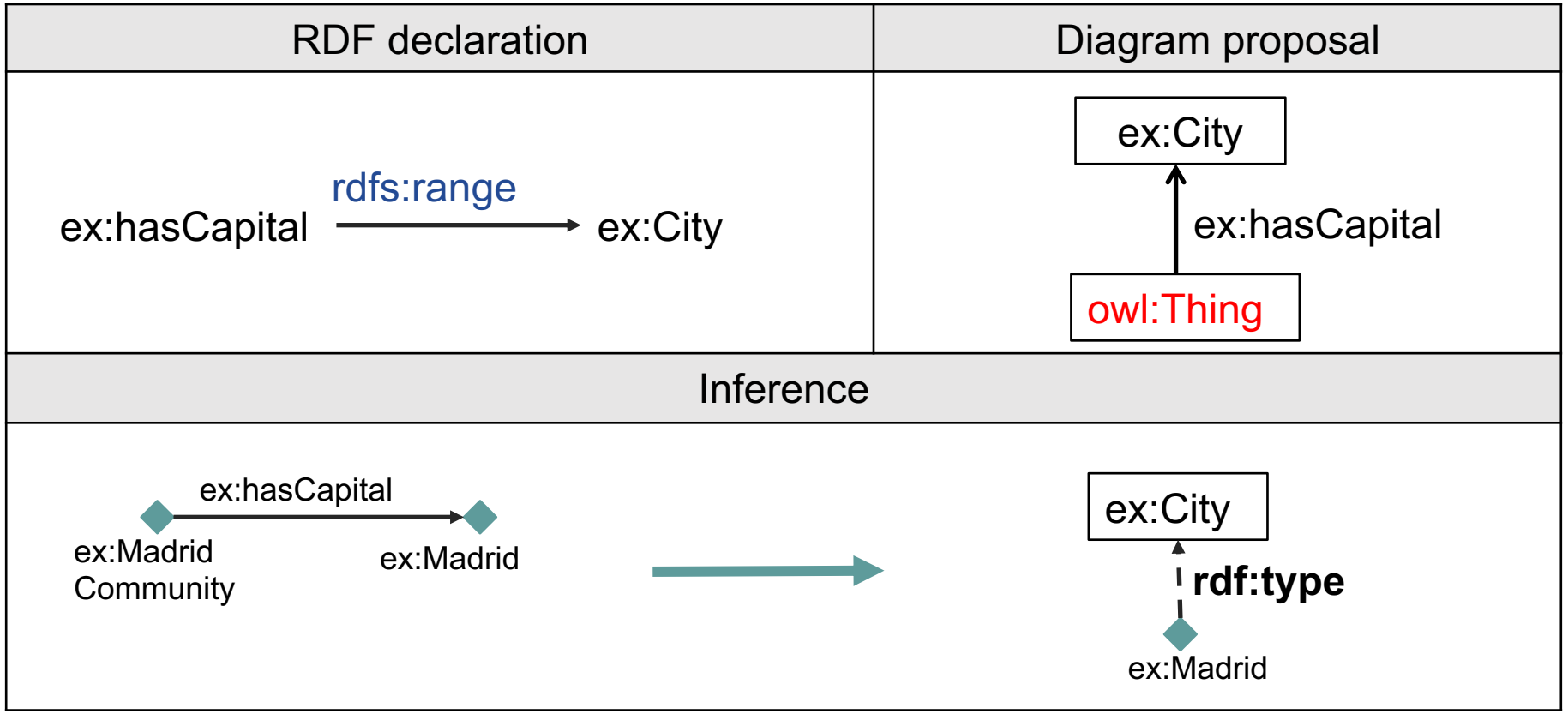
- **rdf:Property**
  - Relation between individuals
  - Relación between individuasl and data values

RDF declaration	Diagram proposal
ex:isPartOf —rdf:type→ rdf:Property Relation expected to be use between individuals  	
ex:hasPopulation —rdf:type→ rdf:Property Example of relation expected to be use between individuals and a data values  	

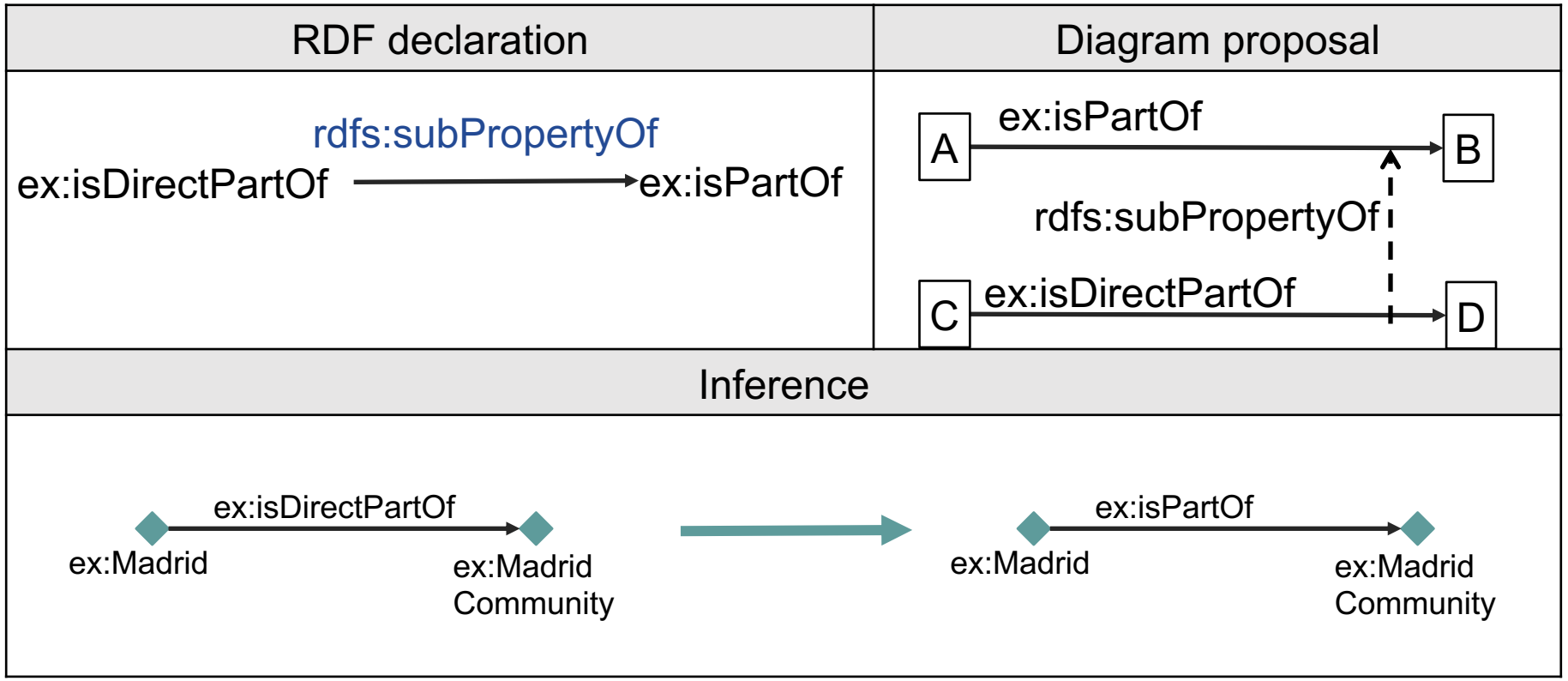
- **rdfs:domain**
  - All individuals that appear as **subject** in a triple with the given property **belongs to** the **class** dfined as **domain** of the property
  - It is valid for relations and attributes



- **rdfs:range**
  - All individuals that appear as **object** in a triple with the given property **belongs to** the **class** dfined as **range** of the property
  - It is valid for clases (applies to relations) or datatypes (applies to attributes)



- **rdfs:subPropertyOf**
  - When there is a property between two elements (two individuals or an individual and a value), the parent property in the hierarchy is also true for the pair of elements
  - Aplicable to *object properties (relations)* and *datatype properties (attributes)*.



Update your model including properties

Identify properties between concepts

Decide whether to add domain and/or range

Individuals

- **rdf:type**
  - Indicate the class or classes of an individual (instance)

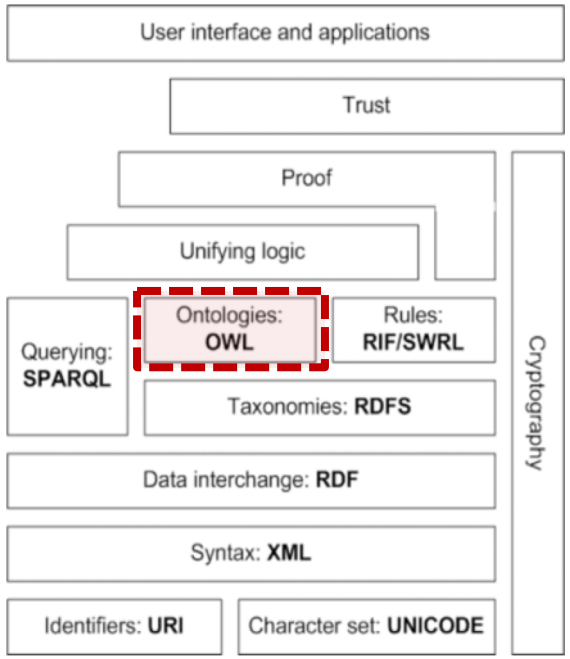
RDF declaration	Diagram proposal
ex:Madrid —rdf:type→ ex:City	ex:City ← **rdf:type** — ex:Madrid ex:City ← **rdf:type** — ex:Madrid

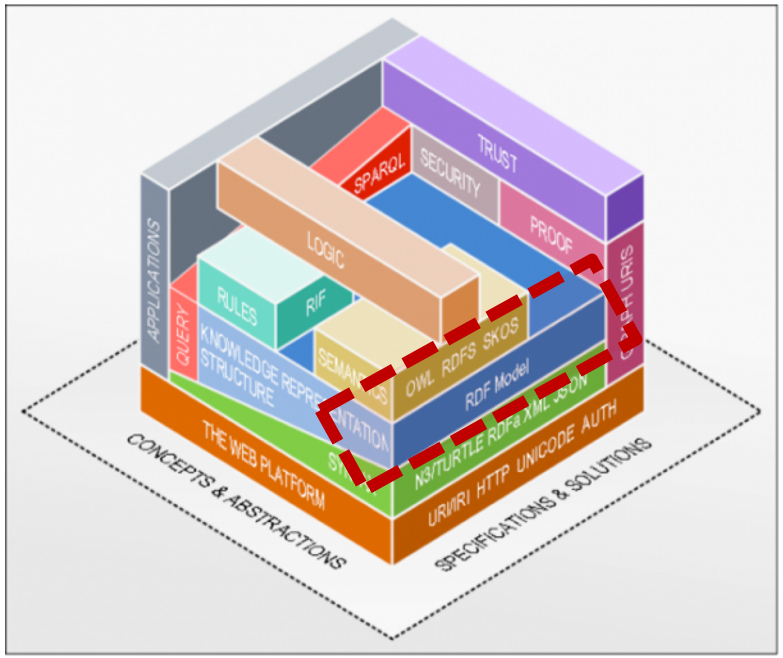
What can’t be modelled from the domain?

- Universities offer subjects
- A subject is delivered in one or more groups
- A subject for a given group is taught by **only 1** professor
  - But in some departments this restriction does not apply
- A subject is offered **only for 1** degree
- A university located at some address
- An address has a street, a number and a postal code
- An address is located in a municipality
- A municipality could have borders with other municipalities.
- A professor could be associate or assistant but **not at the same time**

If A has border with B we might want to infer that B has border with A too

- RDFS is **too weak** to describe resources in sufficient detail
  - No **localised range and domain** constraints
    - Can't say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No **existence/cardinality** constraints
    - Can't say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No **boolean** operators
    - Can't say or, not, etc.
  - No **transitive, inverse or symmetrical** properties
    - Can't say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical



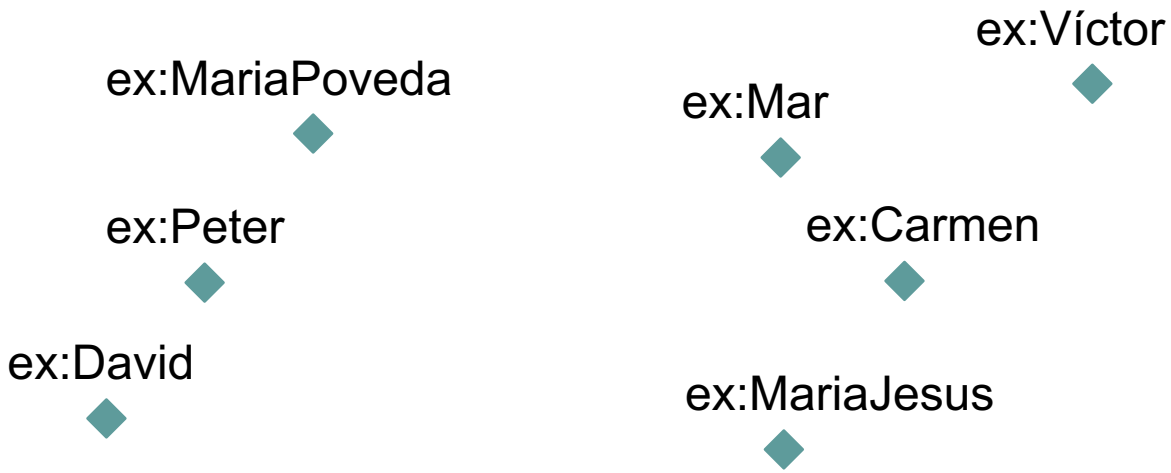


http://w3.org/DesignIssues/diagrams/sweb-stack/2006a.png

- OWL: Web Ontology Language
- Goal
  - To describe the semantic of the information in a machine-readable way
- Based on Description Logics (*DL*)
  - To describiba a domain based on its concepts (classes), roles (relationships) and individuals (instances)
    - Specific languages characterized by the constructs and axioms used to declare knowledge about classes, relations and individuals

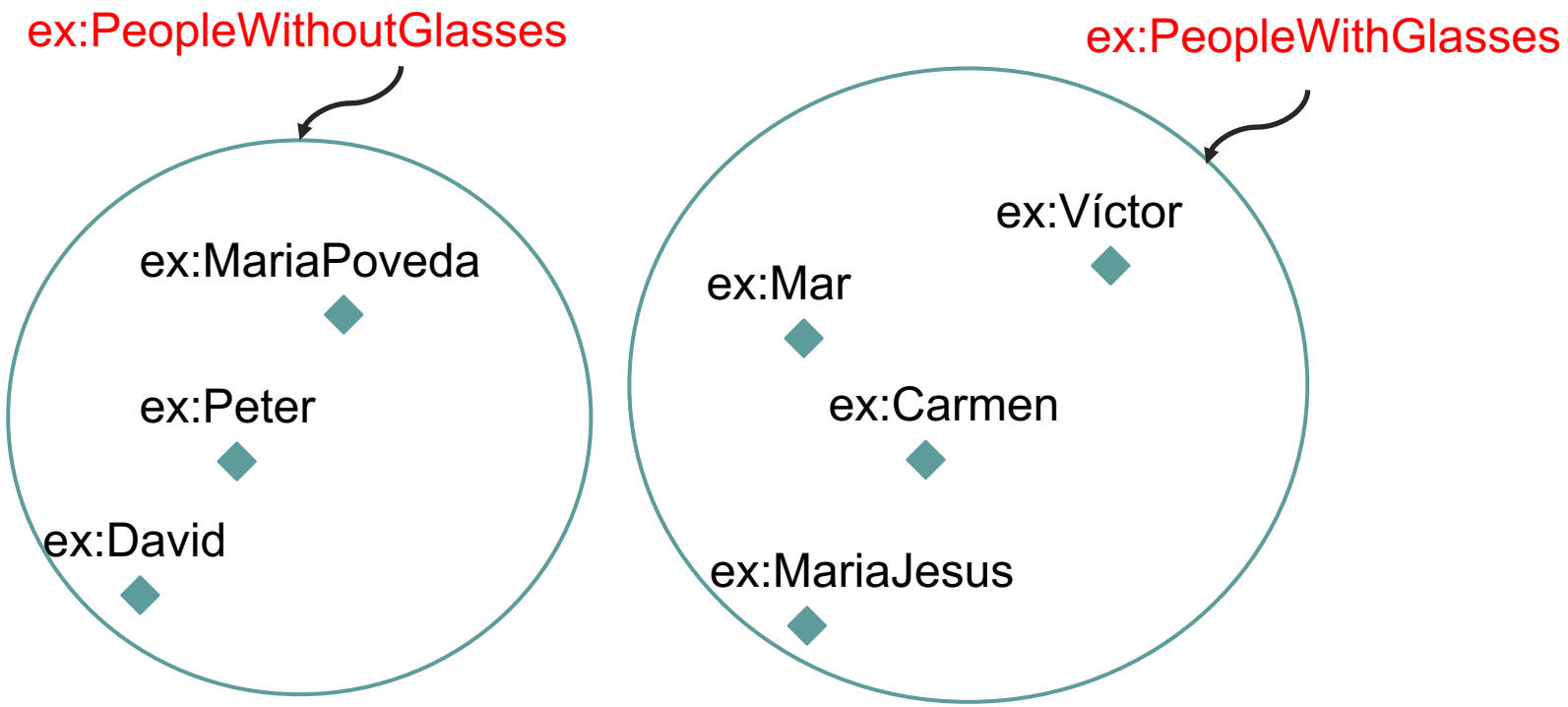
OWL is a FORMAL language

- What is a class?
  - A group of individuals that have a common characteristic



Universe

- What is a class?
  - A group of individuals that have a common characteristic

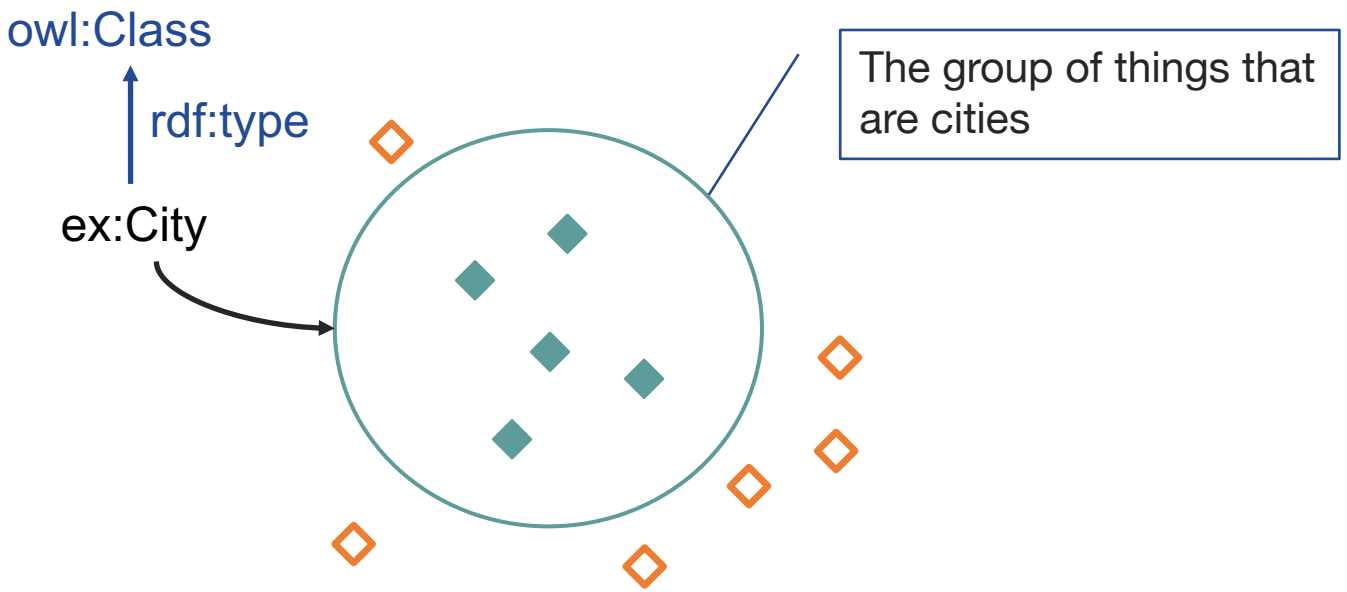


Universe

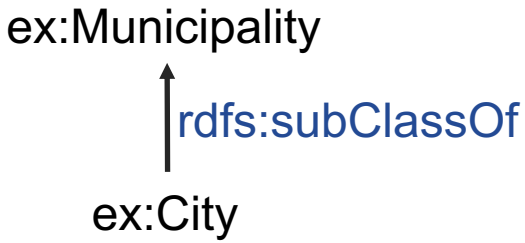
- Remember from set theory:
  - By extension:
    - {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}
  - By intension: “People with glasses in the room”

- … and using OWL?
  - With an identifier: assigning a URI
  - With an exhaustive enumeration of individuals
  - As constraints about a property
  - As union of classes
  - As intersection of classes
  - As complement of a class

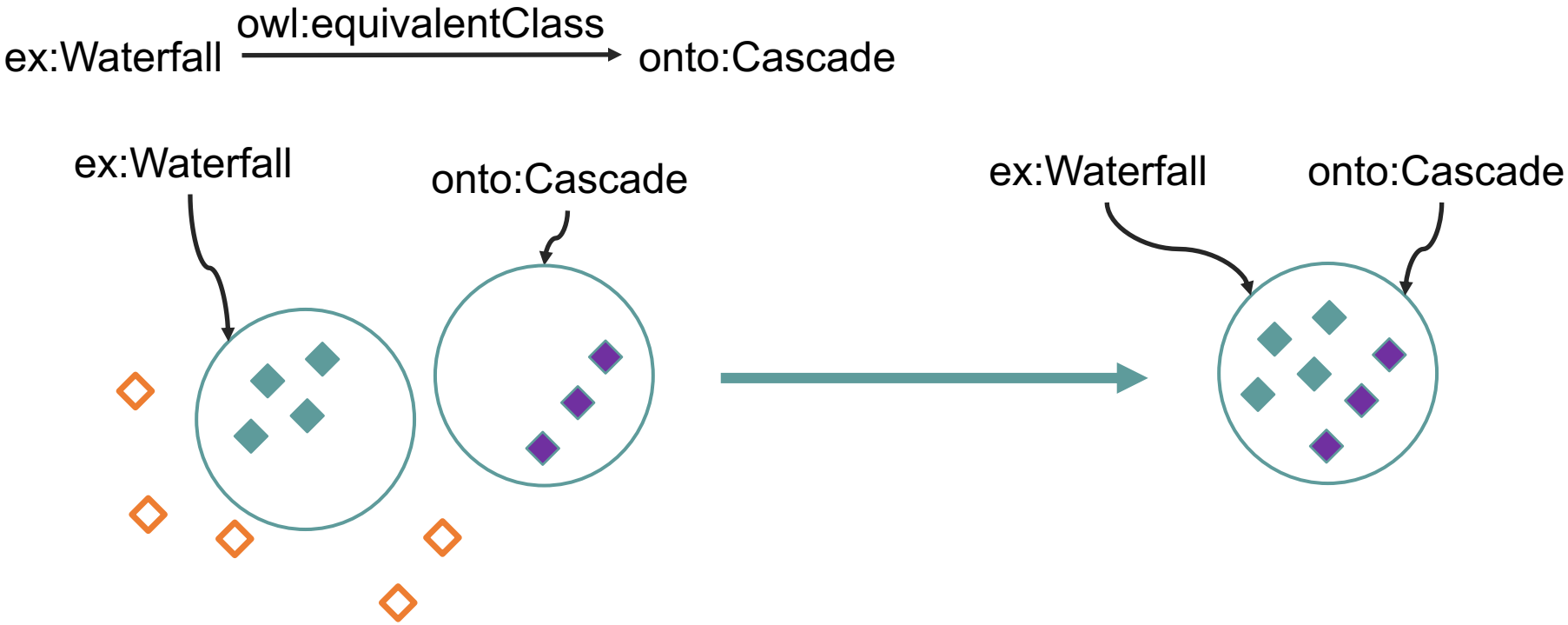
- **owl:Class**
  - Concepts of the domain (generally)
  - **Named**:
    - URI as identifier
  - Not named:
    - Enumeration, constraints over properties, intersection, union, complement.



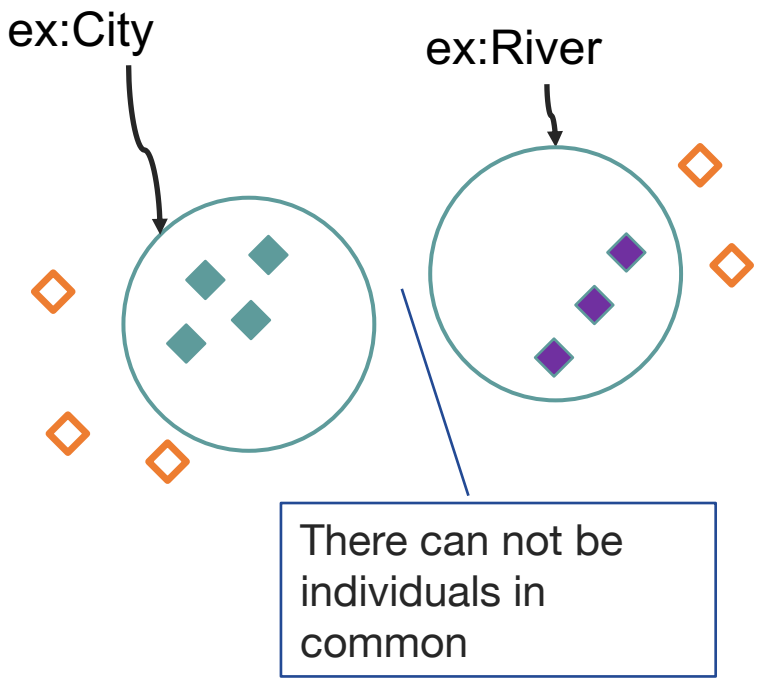
- **rdfs:subClassOf**
  - The individuals belonging to a class also belong to the parent classes in the hierarchy



- **owl:equivalentClass**
  - The two classes contain exactly the same individuals
  - If an individual belongs to a clas it also belongs to the other



- **owl:disjointWith**
  - An individual can not belong to more than one of the involved classes
  - The intersection is the empty set

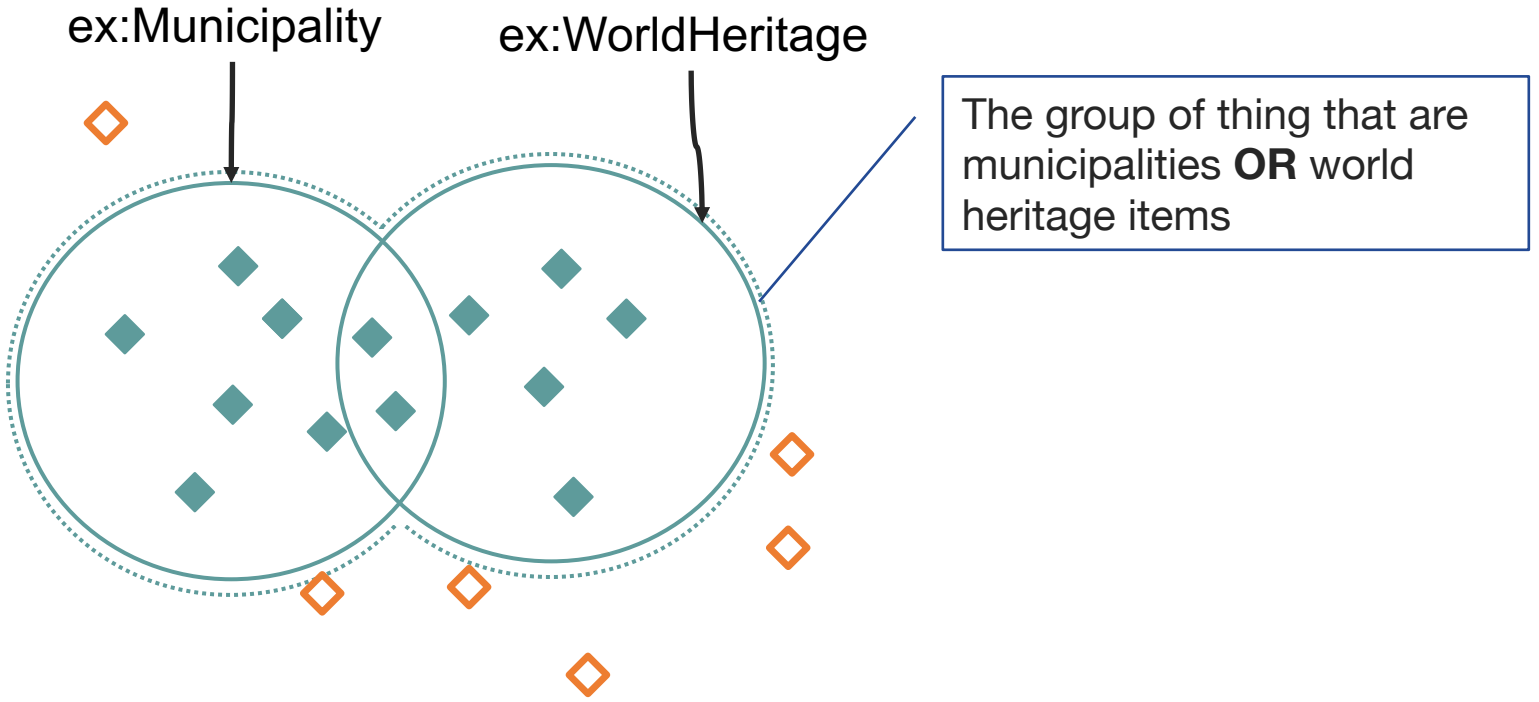


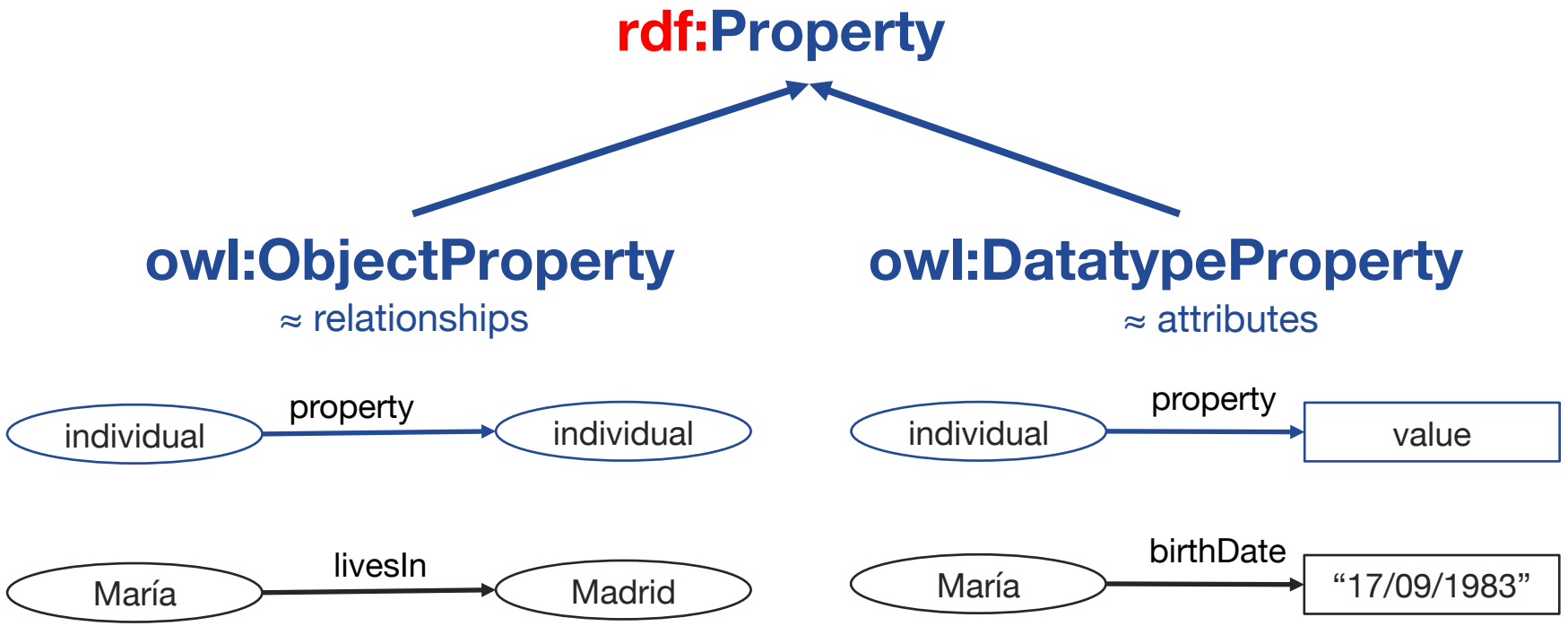
Also…

- owl:unionOf
- owl:intersectionOf
- owl:complementOf
- owl :Thing
- owl :Nothing

- OWL
  - Classes
    - Axioms
  - **Properties**
    - Characteristics
  - Individuals
  - Restrictions

- **owl:unionOf**
  - An individual could belong to one or more of the classes involved
  - **OR** logic

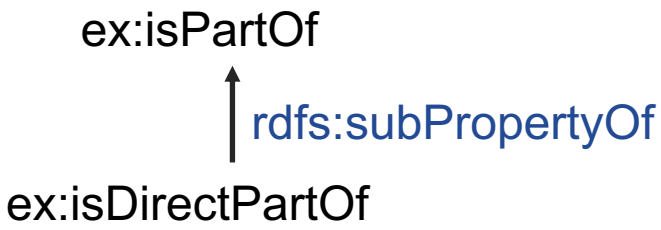



rdf:Property
owl:ObjectProperty
≈ relationships
owl:DatatypeProperty
≈ attributes
individual
property
individual
María
livesIn
Madrid
individual
property
value
María
birthDate
“17/09/1983”

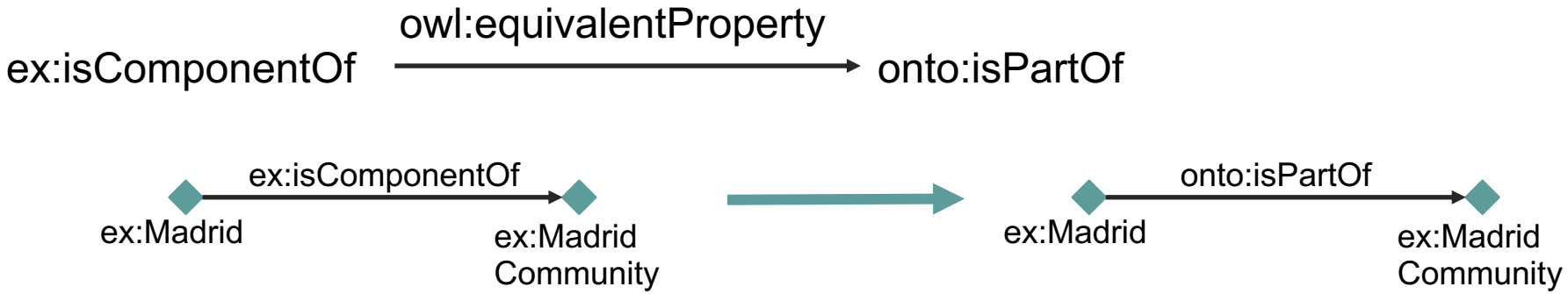
- **rdfs:domain**
  - All individuals that appear as **subject** in a triple with the given property **belongs to** the **class** defined as **domain** of the property
  - It is valid for relations and attributes

- **rdfs:range**
  - All individuals that appear as **object** in a triple with the given property **belongs to** the **class** defined as **range** of the property
  - It is valid for clases (applies to relations) or datatypes (applies to attributes)

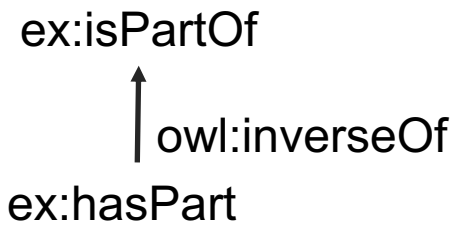
- **rdfs:subPropertyOf**
  - When there is a property between two elements (two individuals or an individual and a value), the parent property in the hierarchy is also true for the pair of elements
  - Aplicable to *object properties (relations)* and *datatype properties (attributes)*.



- **owl:equivalentProperty**
  - When there is a property between two elements (two individuals or an individual and a value), the equivalent property is also true for the pair of elements
  - Aplicable to *object properties (relations)* and *datatype properties (attributes)*.



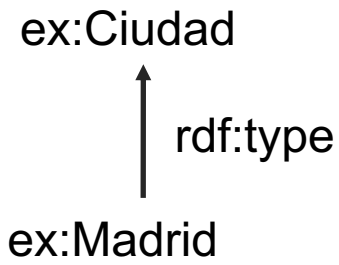
- **owl:inverseOf**
  - When there is a relations between two individuals A (subject) and B (object), the inverse relation is true between the individuals B (subject) and A (object).
  - Aplicable only to *object properties*.

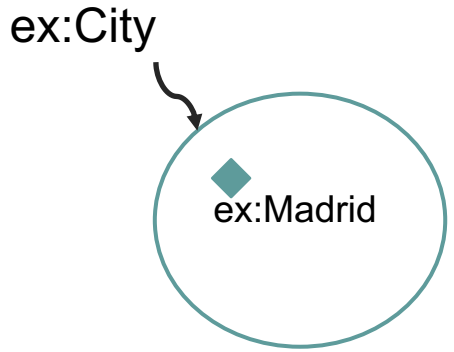


Also

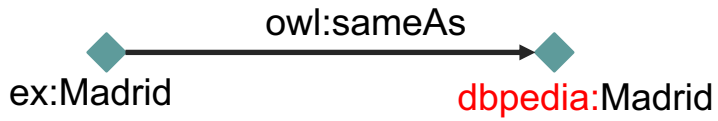
- owl:FunctionalProperty
- owl:InverseFunctionalProperty
- owl:TransitiveProperty
- owl:SymmetricProperty

- **rdf:type**
  - Indicates the class or classes the individual belongs to



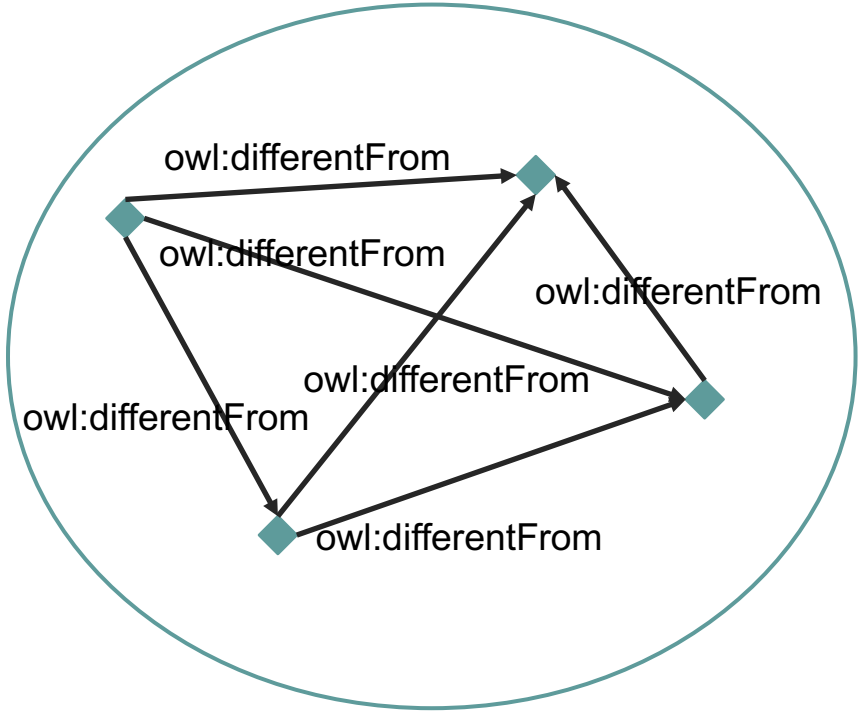


- **owl:sameAs**
  - Declare that two URIs identify the same individual
  - It is defined between instances or individuals



- **owl:differentFrom**
  - Declare that two URIs identify different individuals

- **owl:AllDifferent**
  - Declare that all URIs indicated identigy different individuals
  - Normally used to force the *unique name assumption*



- *Open World Assumption vs Closed World Assumption*
  - OWL follows OWA
  - The lack of evidence about a fact does not imply that the fact is false.
- *Non unique name assumption*
  - Different names do not identified necessarily different individuals.
- OWL 2
  - Additional constructs

- OWL
  - Classes
    - Axioms
  - Properties
    - Characteristics
  - Individuals
  - **Restrictions**
    - If there is still time, just have fun with Protégé

Advanced ontologies and reasoning

María Poveda Villalón, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

[mpoveda@fi.upm.es] @MariaPovedaV

SSoLDAC23