



# Towards a FAIR-by-design ontology development framework

**María Poveda-Villalón**

Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

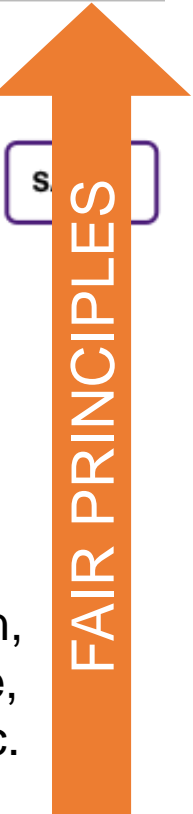
✉ [mpoveda@fi.upm.es](mailto:mpoveda@fi.upm.es)

📅 23<sup>rd</sup> March 2026

📍 AIOTI2026



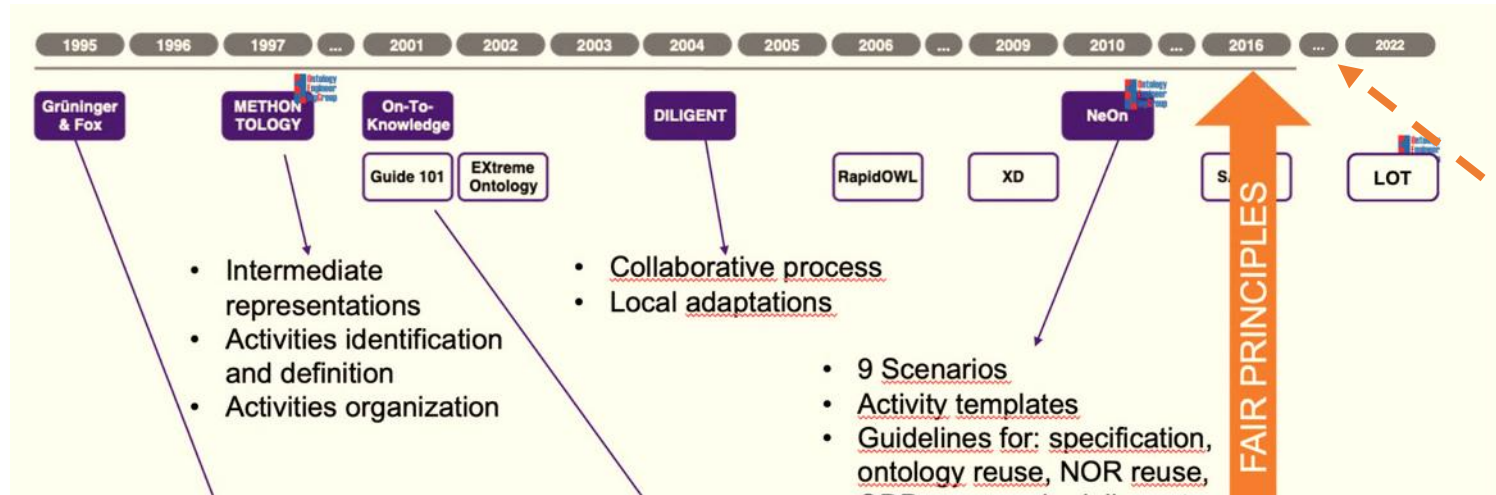
This work was supported by the grant “SOEL: Supporting Ontology Engineering with Large Language Models” PID2023-152703NA-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF/UE”.



Wilkinson, Mark D., et al. "The FAIR Guiding Principles for scientific data management and stewardship." Scientific data 3.1 (2016): 1-9.

O. Development Methodologies

Ontology Development Lightweight Approaches



G&P

- Garijo, D. and Poveda-Villalón, M. (2020) 'Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web', in G. Cota, M. Daquino, and G.L. Pozzato (eds) Studies on the Semantic Web. IOS Press.

Rule

- Cox, S. J., Gonzalez-Beltran, A. N., Magagna, B., & Marinescu, M. (2021) 'Ten simple rules for making a vocabulary FAIR', PLOS Computational Biology, 17(6), p. e1009041.

F-Q

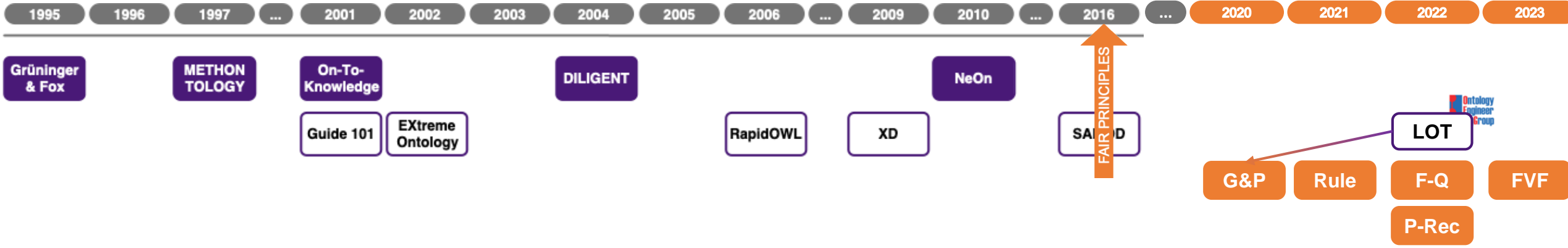
- Amdouni, E., Bouazzouni, S. and Jonquet, C. (2022) 'O'FAIRe makes you an offer: metadata-based automatic FAIRness assessment for ontologies and semantic resources', International Journal of Metadata, Semantics and Ontologies, 16(1), pp. 16–46.

P-Rec

- Le Franc, Y., Bonino, L., Koivula, H., Parland-von Essen, J. & Pergl, R. (2022). 'D2.8 FAIR Semantics Recommendations Third Iteration (V1.0)', FAIRsFAIR, Zenodo.

FVF

- Xu, F. et al. (2023) 'Features of a FAIR vocabulary', Journal of Biomedical Semantics, 14(1), p. 6.



## Linked Open Terms Methodology

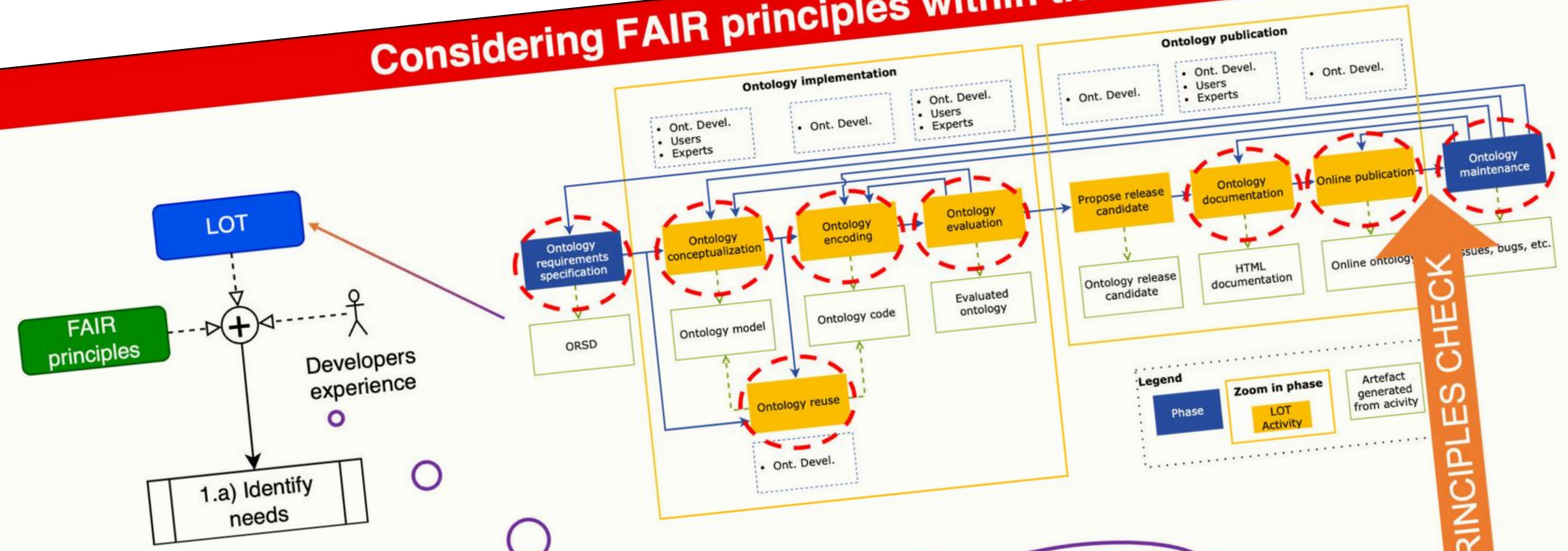


Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., & García-Castro, R. (2022). LOT: An industrial-oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111, 104755. DOI: <https://doi.org/10.1016/j.engappai.2022.104755>

- Towards **lightweight** and agile processes
- Inspiration from software development practices
- Coupling Software and ontology development
- Including Ontology **Publication** Guidelines within the process
- **Best practices** from Semantic Web and **FAIR** principles

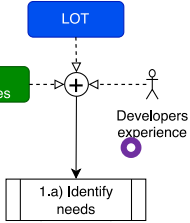
<http://lot.linkeddata.es/>

# Considering FAIR principles within the development process



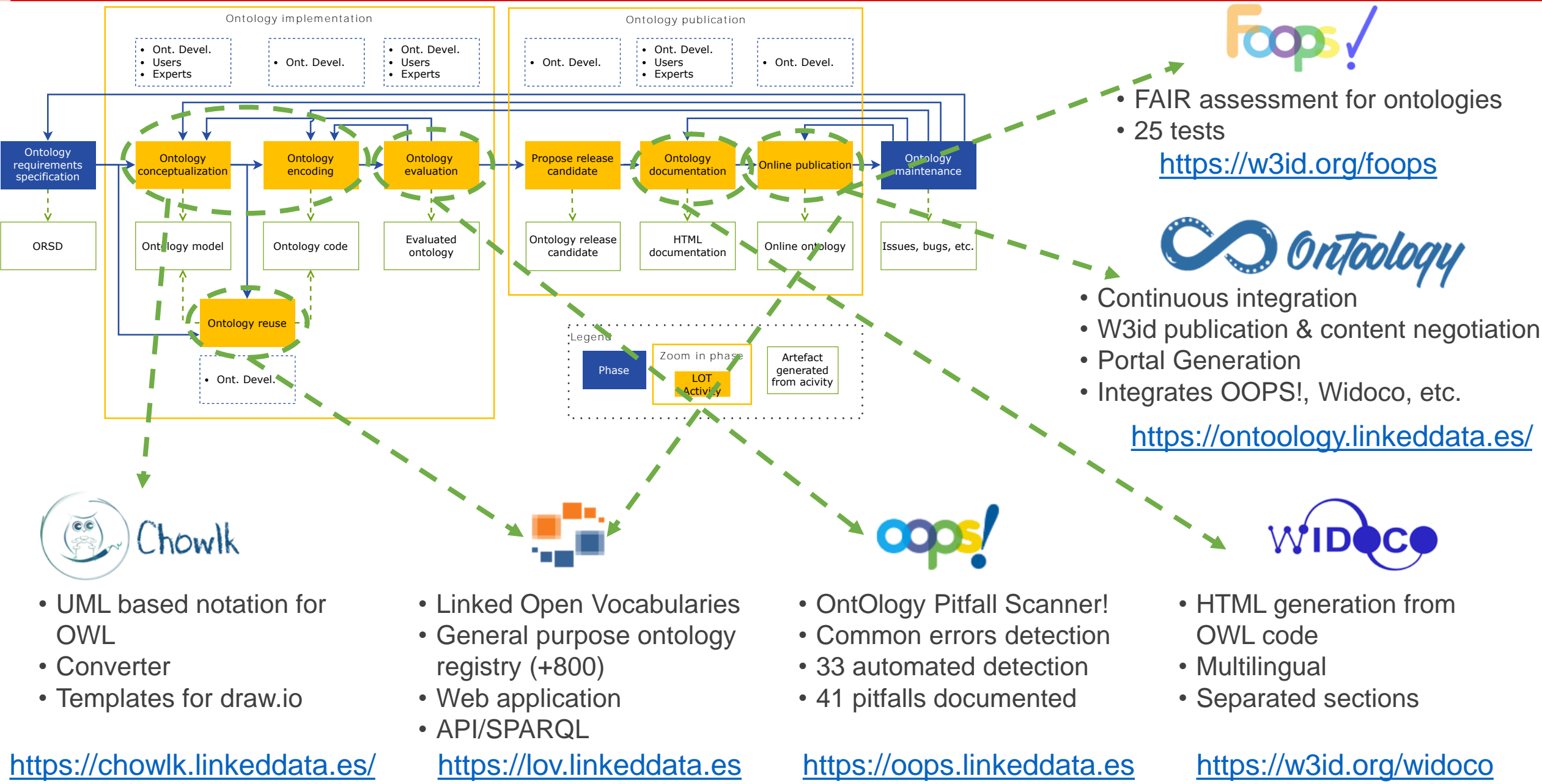
*“Is there something to be done while carrying out the **ACTIVITY\_X** that could affect the FAIRness level of the final ontology with respect to **FAIR\_principle\_Y**?”*

*“Is there something to be done while carrying out the **ACTIVITY\_X** that could affect the FAIRness level of the **final ontology** with respect to **FAIR\_principle\_Y** ?”*



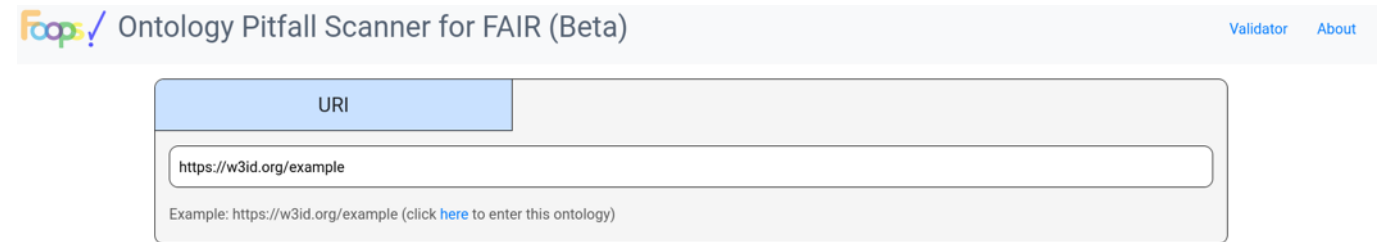
	FAIR Principle	Ontology Requirements Specification	Ontology Reuse	Ontology Conceptualiz.	Ontology Encoding	Ontology Evaluation	Ontology Document.	Ontology Publication	Ontology Maintenance
Findable	F1	X			X		X	X	X
	F2	X	X		X		X		
	F3				X		X	X	
	F4							X	X
Accessible	A1	X	X		X		X	X	
	A1.1	X						X	
	A1.2							X	
	A2				X			X	X
Interoperable	I1	X	X	X	X		X	X	
	I2		X	X	X		X		
	I3		X	X	X		X		X
Reusable	R1	X	X	X	X	X	X		
	R1.1	X	X		X	X	X	X	
	R1.2		X	X	X	X	X	X	
	R1.3	X	X	X	X		X	X	

# My Basic Toolkit for ontology FAIRness boosting






- **Validation service** inspired by OOPS! (Ontology Pitfall Scanner)
- Designed to **guide users (24 tests)**
  - Tests have an explanation
  - Tests indicate potential errors
- **Practical**
  - Based on years of ontology engineering practices at UPM
- Aligned to **FAIR**



<https://w3id.org/foops/>

 Garijo, D., Corcho, O., & Poveda-Villalón, M. (2021). FOOPS!: An Ontology Pitfall Scanner for the FAIR principles. In ISWC (Posters/Demos/Industry). Best demo award.

The screenshot displays the FOOPS! report interface. At the top, the 'Ontology metadata summary' includes fields for Title ('The example ontology'), URI ('https://w3id.org/example'), and License ('http://creativecommons.org/licenses/by/2.0/'). Below this, a circular progress indicator shows a 78% overall score. A diamond-shaped chart breaks down the score by category: Findable (8/9), Reusable (6.75/9), Accessible (2/3), and Interoperable (2/3). The main report is organized into expandable sections: Findable, Accessible, and Interoperable. The Interoperable section is expanded, showing two checks: I1: (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation, and I2: (meta)data use vocabularies that follow FAIR principles. Under I2, there are two sub-checks: VOC1: Vocabulary reuse (metadata) with 100% coverage, and VOC2: Vocabulary reuse with 0% coverage. A description and explanation are provided for VOC2.

Ontology metadata summary

78%

Findable (8/9)  
Reusable (6.75/9)  
Accessible (2/3)  
Interoperable (2/3)

Findable

Accessible

Interoperable

I1: (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation

RDF1: RDF Availability 100%

I2: (meta)data use vocabularies that follow FAIR principles

VOC1: Vocabulary reuse (metadata) 100%

VOC2: Vocabulary reuse 0%

Description: This check verifies if the ontology imports/extends other vocabularies (besides RDF, OWL and RDFS)

Explanation: Could not find any imported/reused vocabularies

FAIRness coverage by category

FAIRness overall score.

**Note:** this may be a **quality indicator**, but there is no defined threshold for FAIRness.

FAIR Category

Check

Check coverage

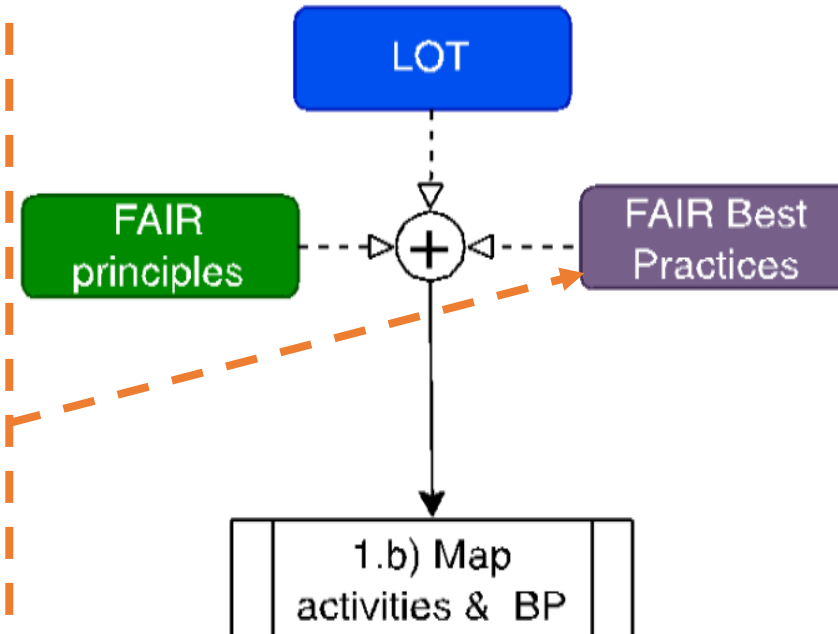
Check description

Check explanation

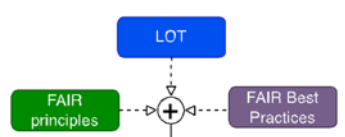
## Conclusions and Future Work

- **Currently:** checking/improving the FAIRness level of ontologies is done once an ontology is **built** and **published** (e.g., using ~~FOOPS!~~ or ~~O'FAIRe~~).
  - However, increasing the FAIRness level of an ontology should be addressed along its **whole development life-cycle** (e.g., carefully documenting design decisions and requirements, including ontology metadata in the conceptualization phase, etc.)
- Identified tools that help ontology FAIRness
- Need: Increase tools support for FAIR compliance ontologies, for example facilitating metadata updates and maintenance

- ✦ **G&P-X:** Garijo, D. and Poveda-Villalón, M. (2020) 'Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web', in G. Cota, M. Daquino, and G.L. Pozzato (eds) Studies on the Semantic Web. IOS Press.
- ✦ **Rule-X:** Cox, S. J., Gonzalez-Beltran, A. N., Magagna, B., & Marinescu, M. (2021) 'Ten simple rules for making a vocabulary FAIR', PLOS Computational Biology, 17(6), p. e1009041.
- ✦ **FX-QY:** Amdouni, E., Bouazzouni, S. and Jonquet, C. (2022) 'O'FAIRe makes you an offer: metadata-based automatic FAIRness assessment for ontologies and semantic resources', International Journal of Metadata, Semantics and Ontologies, 16(1), pp. 16–46.
- ✦ **P-rec-X:** Le Franc, Y., Bonino, L., Koivula, H., Parland-von Essen, J. & Pergl, R. (2022). 'D2.8 FAIR Semantics Recommendations Third Iteration (V1.0),' FAIRsFAIR.
- ✦ **FVF-X:** Xu, F. et al. (2023) 'Features of a FAIR vocabulary', Journal of Biomedical Semantics, 14(1), p. 6.



# Methodology – Map activities and BP



FAIR Principle	Ontology Requirements Specification	Ontology Reuse	Ontology Conceptualiz.	Ontology Encoding	Ontology Evaluation	Ontology Document
F1				Y		Y
F2				Y		Y
F3				Y		Y
F4						
A1		Y		Y		Y
A1.1						
A1.2						Y
A2				Y		
I1			Y	Y		
I2		Y	Y	Y		
I3		Y		Y		Y
R1		Y	Y	Y	Y	Y
R1.1		Y		Y		Y
R1.2		Y	Y	Y		Y
R1.3			Y	Y		Y

**F2. data are described with rich metadata (defined by R1 below)**

- FVF-2: Vocabularies and their terms have rich metadata.
- G&P-6: Generate ontology metadata (includes mandatory and optional properties)
- P-Rec3: A common minimum metadata schema must be used to describe semantic artefacts and their content
- Rule-7: Add vocabulary metadata

**A1. (meta)data are retrievable by their identifier using a standardized communications protocol**

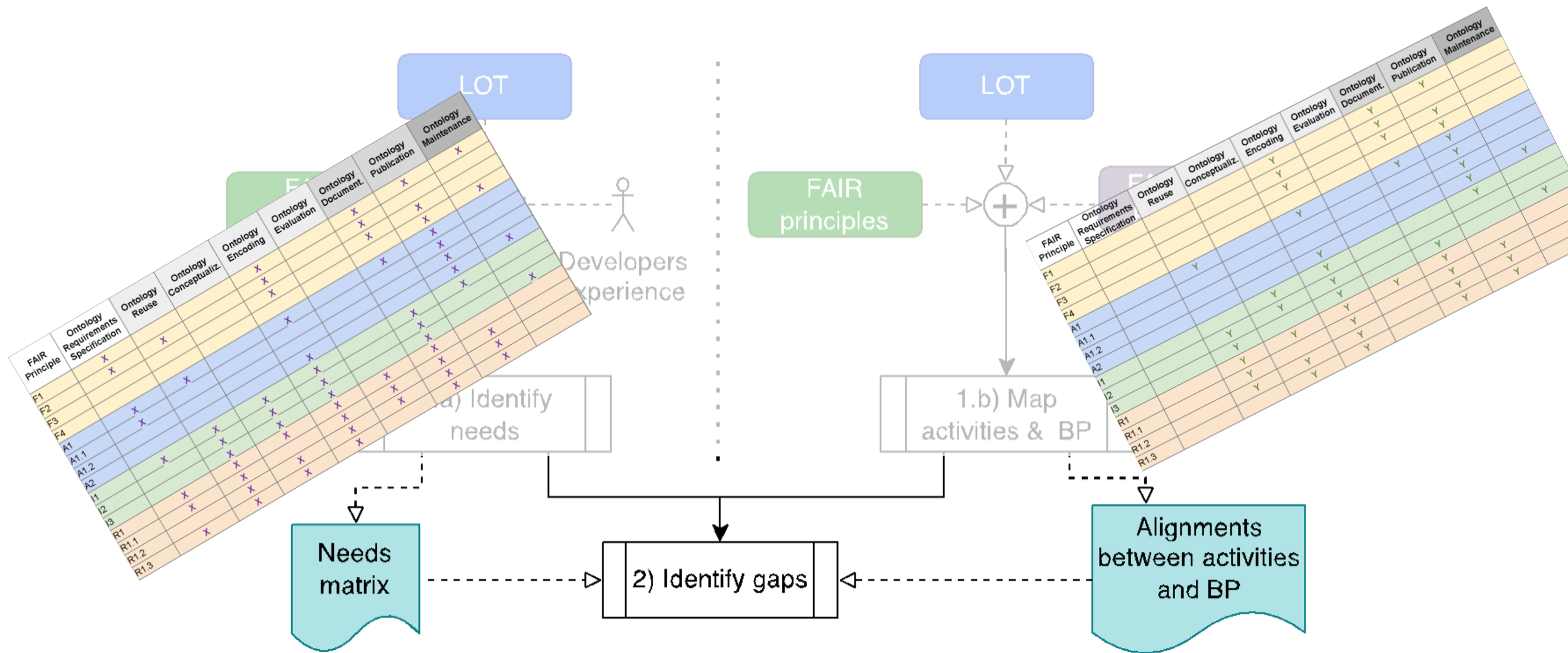
- G&P-1: Design ontology name and prefix
- G&P-2: Decide between hash or slash URIs
- G&P-3: Decide whether to use opaque URIs
- G&P-4: Define an ontology versioning strategy

**I2. (meta)data use vocabularies that follow FAIR principles**

- I2Q6: Does the ontology provide information about the relation to or influence of other FAIR vocabularies?
- P-Rec14: Standard vocabularies should be used to describe semantic artefacts

**R1.2. (meta)data are associated with detailed provenance**

- G&P-6: Generate ontology metadata
- G&P-8: Generate diagrams



Needs matrix

2) Identify gaps

Alignments between activities and BP

# Methodology – Identify Gaps

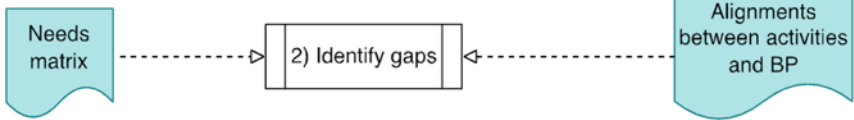
Principle	Ontology Requirements Specification		Ontology Reuse		Ontology Conceptualiz.		Ontology Encoding		Ontology Evaluation		Ontology Document.		Ontology Publication		Ontology Maintenance	
	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref
F1	X	GAP					X	Y			X	Y	X	Y	X	GAP
F2	X	GAP	X	GAP			X	Y			X	Y				
F3							X	Y			X	Y	X	Y		
F4													X	Y	X	GAP
A1	X	GAP	X	Y			X	Y			X	Y	X	Y		
A1.1	X	GAP											X	Y		
A1.2													X	Y		
A2							X	Y					X	Y	X	Y
I1	X	GAP	X	Y	X	Y	X	Y			X	GAP	X	Y		
I2			X	Y	X	Y	X	Y			X	GAP				
I3			X	Y	X	GAP	X	Y			X	Y			X	Y
R1	X	GAP	X	Y	X	Y	X	Y	X	Y	X	Y				
R1.1	X	GAP	X	Y			X	Y	X	GAP	X	Y	X	Y		
R1.2			X	Y	X	Y	X	Y	X	GAP	X	Y	X	Y		
R1.3	X	GAP	X	GAP	X	Y	X	Y			X	Y	X	Y		

High availability of guidelines

Principle	Ontology Requirements Specification		Ontology Reuse		Ontology Conceptualiz.		Ontology Encoding		Ontology Evaluation		Ontology Document.		Ontology Publication		Ontology Maintenance	
	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref
F1	X	GAP					X	Y			X	Y	X	Y	X	GAP
F2	X	GAP	X	GAP			X	Y			X	Y				
F3							X	Y			X	Y	X	Y		
F4													X	Y	X	GAP
A1	X	GAP	X	Y			X	Y			X	Y	X	Y		
A1.1	X	GAP											X	Y		
A1.2													X	Y		
A2							X	Y					X	Y	X	Y
I1	X	GAP	X	Y	X	Y	X	Y			X	GAP	X	Y		
I2			X	Y	X	Y	X	Y			X	GAP				
I3			X	Y	X	GAP	X	Y			X	Y			X	Y
R1	X	GAP	X	Y	X	Y	X	Y	X	Y	X	Y				
R1.1	X	GAP	X	Y			X	Y	X	GAP	X	Y	X	Y		
R1.2			X	Y	X	Y	X	Y	X	GAP	X	Y	X	Y		
R1.3	X	GAP	X	GAP	X	Y	X	Y			X	Y	X	Y		

Partial coverage

Principle	Ontology Requirements Specification		Ontology Reuse		Ontology Conceptualiz.		Ontology Encoding		Ontology Evaluation		Ontology Document.		Ontology Publication		Ontology Maintenance	
	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref
F1	X	GAP					X	Y			X	Y	X	Y	X	GAP
F2	X	GAP	X	GAP			X	Y			X	Y				
F3							X	Y			X	Y	X	Y		
F4													X	Y	X	GAP
A1	X	GAP	X	Y			X	Y			X	Y	X	Y		
A1.1	X	GAP											X	Y		
A1.2													X	Y		
A2							X	Y					X	Y	X	Y
I1	X	GAP	X	Y	X	Y	X	Y			X	GAP	X	Y		
I2			X	Y	X	Y	X	Y			X	GAP				
I3			X	Y	X	GAP	X	Y			X	Y			X	Y
R1	X	GAP	X	Y	X	Y	X	Y	X	Y	X	Y				
R1.1	X	GAP	X	Y			X	Y	X	GAP	X	Y	X	Y		
R1.2			X	Y	X	Y	X	Y	X	GAP	X	Y	X	Y		
R1.3	X	GAP	X	GAP	X	Y	X	Y			X	Y	X	Y		

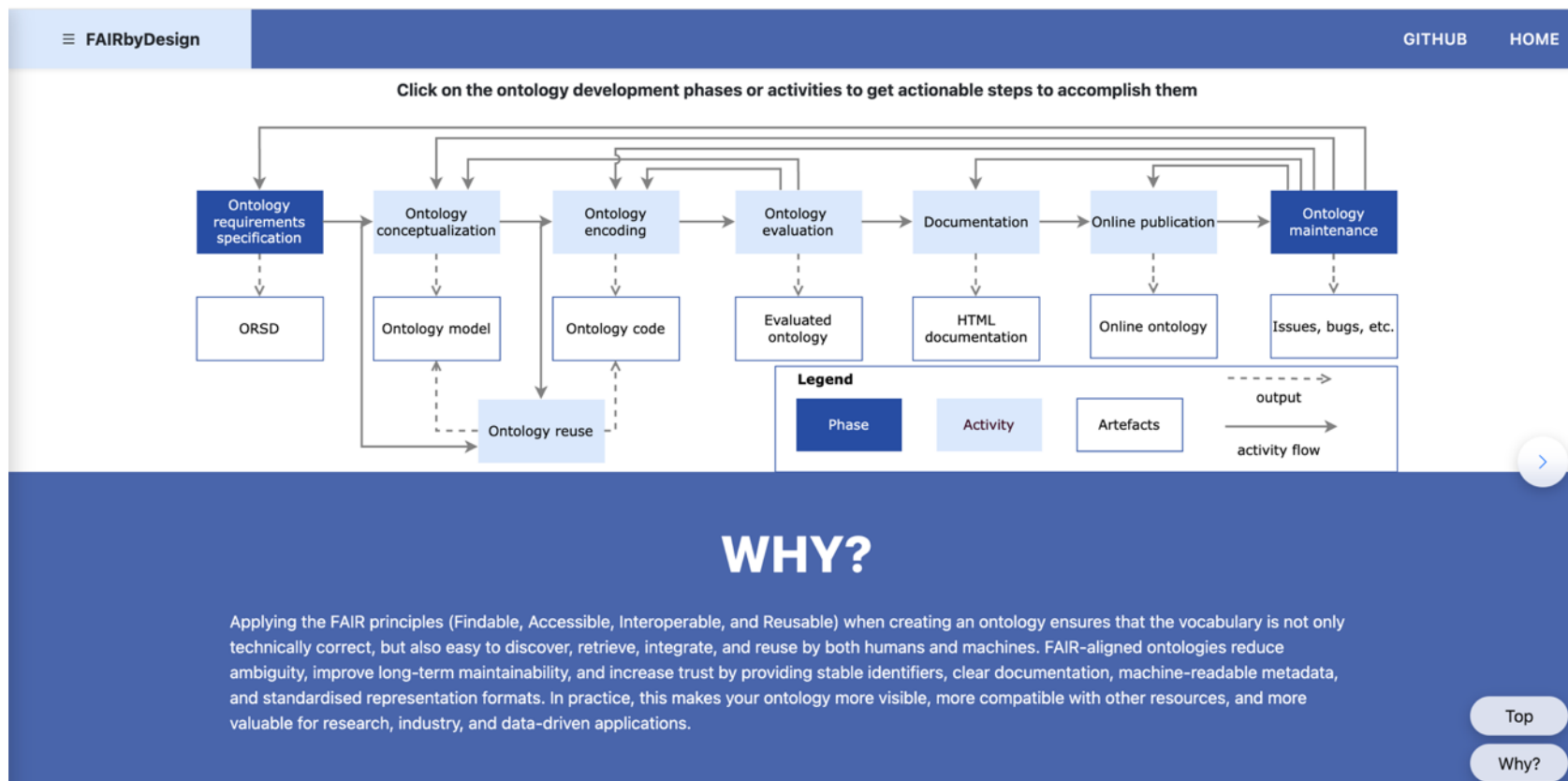


Low coverage

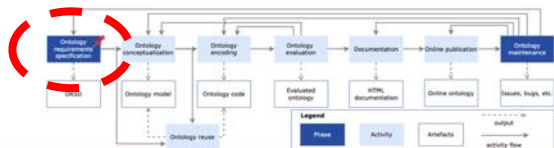
Principle	Ontology Requirements Specification		Ontology Reuse		Ontology Conceptualiz.		Ontology Encoding		Ontology Evaluation		Ontology Document.		Ontology Publication		Ontology Maintenance	
	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref
F1	X	GAP					X	Y			X	Y	X	Y	X	GAP
F2	X	GAP	X	GAP			X	Y			X	Y				
F3							X	Y			X	Y	X	Y		
F4													X	Y	X	GAP
A1	X	GAP	X	Y			X	Y			X	Y	X	Y		
A1.1	X	GAP											X	Y		
A1.2													X	Y		
A2							X	Y					X	Y	X	Y
I1	X	GAP	X	Y	X	Y	X	Y			X	GAP	X	Y		
I2			X	Y	X	Y	X	Y			X	GAP				
I3			X	Y	X	GAP	X	Y			X	Y			X	Y
R1	X	GAP	X	Y	X	Y	X	Y	X	Y	X	Y				
R1.1	X	GAP	X	Y			X	Y	X	GAP	X	Y	X	Y		
R1.2			X	Y	X	Y	X	Y	X	GAP	X	Y	X	Y		
R1.3	X	GAP	X	GAP	X	Y	X	Y			X	Y	X	Y		

<https://w3id.org/sa-fair-by-design/>

- **Actionable** LOT FAIRbyDesign methodology
- Lightweight **checks** and **tools** integrated per activity



Click on the ontology development phases or activities to get actionable steps to accomplish them



## Ontology requirements specification

The aim of the requirements specification phase is to state why the ontology is being built and to identify and define the requirements the ontology should fulfill.

### Steps

Top

0% completed Why?

Principles

Mark All Reset

#### 1. Provide identifiers to functional requirements

Checklist

Assign identifiers to functional requirements (competency questions or other forms).

#### 2. Provide links to functional requirements

Checklist

Provide links to functional requirements (competency questions or other forms) either in a compiled version or separated in different files.

#### 3. Provide versioning information

Checklist

Provide versioning information about the requirements and the ontology versions.

#### 4. Provide requirements metadata

Checklist

Provide metadata information about requirements (e.g., requirements identifier, author, creation date, version of the ORSD, license of the ORSD, keywords, produced ontology, etc.).

#### 5. Use HTTP URIs for requirements

Checklist

Use HTTP URIs to identify requirements (or sets) and retrieve them.

#### 6. Provide requirements in an open format

Checklist

Provide requirements at least in an open format file as CSV or text. You can use the Ontology Requirement Specification Document templates following the NeOn and LOT methodology, for example: LOT resources templates for ORSD.

## Low coverage

	Ontology Requirements Specification		Ontology Reuse		Ontology Conceptualiz.		Ontology Encoding		Ontology Evaluation		Ontology Document.		Ontology Publication		Ontology Maintenance	
Principle	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref	Need	Ref
F1	X	GAP					X	Y			X	Y	X	Y	X	GAP
F2	X	GAP	X	GAP			X	Y			X	Y				
F3							X	Y			X	Y				
F4													X	Y	X	GAP
A1	X	GAP	X	Y			X	Y			X	Y	X	Y		
A1.1	X	GAP											X	Y		
A1.2													X	Y		
A2							X	Y					X	Y	X	Y
I1	X	GAP	X	Y	X	Y	X	Y			X	GAP	X	Y		
I2			X	Y	X	Y	X	Y			X	GAP				
I3			X	Y	X	GAP	X	Y			X	Y			X	Y
R1	X	GAP	X	Y	X	Y	X	Y	X	Y	X	Y				
R1.1	X	GAP	X	Y			X	Y	X	GAP	X	Y	X	Y		
R1.2			X	Y	X	Y	X	Y	X	GAP	X	Y	X	Y		
R1.3	X	GAP	X	GAP	X	Y	X	Y			X	Y	X	Y		

## Why?

Ontological requirements play a crucial role in ontology development projects defining the needs that the intended model should fulfill. Even though they are rarely published together with the resulting ontologies, ontology requirements could be valuable for potential users to understand the ontology. In addition, ontological requirements can be considered not only as metadata about the resulting ontology but also as provenance information. For these reasons, ontology requirements should be treated as first class citizens and key resources during the ontology development process, applying the FAIR Principles to them when possible.

## Resources

- ORSD (Ontology Requirements Specification Document)
- List of Functional Requirements (Competency Questions or natural language affirmative sentences)
- (optional) Functional Requirements Formalization

## Roles

- Ontology development team
- Ontology users
- Domain experts

## PRINCIPLES

Findable

Accessible

Interoperable

Reusable

### Findable

FBD-Req1: Provide identifiers to functional requirements (competency questions or other forms).

FBD-Req2: Provide links to functional requirements (competency questions or other forms) either in a compiled version or separated in different files.

FBD-Req3: Provide versioning information about the requirements and the ontology versions.

FBD-Req4: Provide metadata information about requirements (e.g., authors, projects related, modification dates, status).

### Accessible

FBD-Req5: Use HTTP URIs to identify requirements (or sets) and retrieve them.

### Interoperable

FBD-Req6: Provide requirements at least in an open format file as csv or text.

FBD-Req7: Optimal: Provide RDF description of the requirements retrievable from their URIs (see: <http://w3id.org/def/ontoreq>).

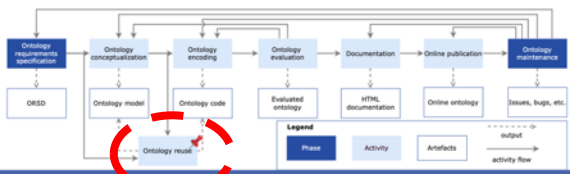
### Reusable

FBD-Req8: Requirements are richly described as instances of formal objects from a relevant metadata vocabulary such as MOD or OntoReq. Recommended metadata fields are: requirements identifier, author, creation date, version of the ORSD, license of the ORSD, keywords, produced ontology

FBD-Req9: Define a usage license for the requirements.

FBD-Req10: Use available Ontology Requirement Specification Document templates following the NeOn and LOT methodology, for example: <https://github.com/oeg-upm/LOT-resources/tree/master/templates%20for%20ORSRD>.

Click on the ontology development phases or activities to get actionable steps to accomplish them



## Ontology reuse

This activity refers to using and employing an existing ontology or vocabulary when developing a new ontology

FAIRbyDesign

GITHUB HOME

### Steps

Run all tests

#### 1. Ontology reuses existing vocabularies for metadata annotations

Enter ontology URI

Submit

#### 2. Ontology imports or reuses well established vocabularies

Enter ontology URI

Submit

#### 3. Check accessibility of reused ontologies/terms

Checklist

Check that the reused ontologies or terms are accessible as any other objects within the ontology.

Top

Why?

Principles

#### 4. Check metadata of reused terms

Checklist

Check metadata of reused terms.

#### 5. Check licenses and repurposing permissions

Checklist

Check license of the reused vocabularies and whether they allow repurposing.

#### 6. Use standard languages/primitives for reuse

Checklist

Use standard languages for implementing the reuse, for example OWL primitives (`owl:equivalentClass`, `owl:equivalentProperty`, etc.), RDF(S) primitives (`rdfs:subClassOf`, `rdfs:subPropertyOf`), SKOS properties (`skos:broader`, `skos:narrower`, etc.), SSSOM, etc.

Check reuse in ontology metadata and content



Check list for developers

### Why?

During the ontology reuse activity, candidate ontologies to be reused are analyzed and decisions about which ontologies will be reused (i.e. imported, extended, specialized, generalized, etc.) and how are made. As the reused parts become part of the ontology being built, they should be treated as final outputs and FAIR Principles should be applied to them. In addition, thus the team reusing existing ontologies shall pay attention to favour FAIR components.

### Resources

- Ontology code
- Ontology conceptualization
- Ontology requirements

### Roles

- Ontology development team
- Domain experts

### PRINCIPLES

- Findable
- Accessible
- Interoperable
- Reusable

#### Findable

FBD-Reuse1: Provide metadata about which ontologies are being reused (understanding reuse as an extension, import, specialisation or generalisation of an ontology).  
FBD-Reuse2: Verify if the reused ontology should be referenced by including their corresponding versioning information (in case the version IRI is provided by the original ontology). [Additional reference: MIREOT <https://doi.org/10.1038/npre.2009.3576.1>]

#### Accessible

A1Q1: Do the ontology URI and other identifiers, if they exist, resolve to the ontology?  
A1Q2: Does the ontology URI (if metadata are included in the ontology file) or the external metadata URI resolve to the metadata record?  
A1Q3: Do the ontology URI and the external metadata URI (if the metadata are not included in the ontology file), support content negotiation?  
FBD-Reuse3: The reused ontologies or terms shall be accessible as any other objects within the ontology (all Accessibility principles affected).

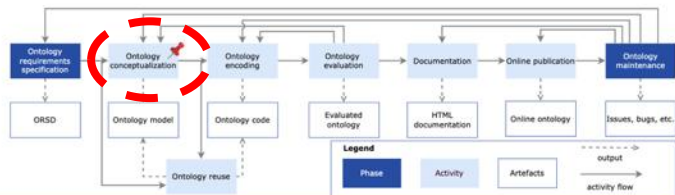
#### Interoperable

I1Q1: What is the representation language used for the ontology and ontology metadata?  
I1Q2: Is the representation language used in a W3C Recommendation?  
FBD-Reuse4: Reuse ontologies and terms are (or are coming from) resources implemented in standard languages.  
FBD-Reuse5: Use standard languages for implementing the reuse, for example OWL primitives (`owl:equivalentClass`, `owl:equivalentProperty`, etc.), RDF(S) primitives (`rdfs:subClassOf`, `rdfs:subPropertyOf`), SKOS properties (`skos:broader`, `skos:narrower`, etc.), SSSOM, etc.  
I2Q1: Does the ontology import other FAIR vocabularies?  
I2Q2: Does the ontology reuse terms from other FAIR vocabularies (URIs)?  
I2Q3: If yes, does it include the minimum information for those terms?  
I2Q3: If yes, does it include the minimum information for those terms?  
I2Q5: If yes, are those alignments well represented and to unambiguous entities? If yes, are those alignments curated?  
I2Q7: Does the ontology reuse standard and FAIR metadata vocabularies to describe its metadata?  
FBD-Reuse6: Assess candidate ontologies/vocabularies FAIRness level. Check the FAIRness level within a repository <https://wheat.agroportal.lirmm.fr/> or by stand alone applications (e.g. FOOPSI <https://w3id.org/foopsi/>)  
FVF-8: Vocabularies and terms use qualified references to other vocabularies.

#### Reusable

I2Q3: If yes, does it include the minimum information for those terms?  
Rule-2: Verify that the legacy-vocabulary license allows repurposing, and agree on the license for the FAIR vocabulary.  
FBD-Reuse7: Identify the licenses of the reused vocabularies and analyse their compatibility.  
FBD-Reuse8: Provide a license for the ontology being built compatible with the licenses of the reused ontologies.  
P-Rec15: Provenance information regarding the reuse of components from third-party semantic artefacts should be made explicit.  
FBD-Reuse9: Use standard languages for implementing the reuse, for example OWL primitives (`owl:equivalentClass`, `owl:equivalentProperty`, etc.), RDF(S) primitives (`rdfs:subClassOf`, `rdfs:subPropertyOf`), SKOS properties (`skos:broader`, `skos:narrower`, etc.), SSSOM, etc.

Click on the ontology development phases or activities to get actionable steps to accomplish them



## Ontology conceptualization

During the ontology conceptualization, the domain knowledge obtained from the ORSD document is organised and structured into a model by the ontology developers in collaboration with domain experts and ontology users. Usually, diagrams are generated to sketch the ontology hierarchies and main relations between concepts. It is also possible to generate logic based representations prior to the actual implementation. During this activity notes and documentation related to modelling decisions could be generated.

### Steps

Top

Why?

Principles

#### 1. Create a diagram

Create an initial conceptual diagram of the ontology (main classes, key relations, and constraints) to agree on scope and structure before implementation. A practical way to do this is to use **Chowlk**, you can use it to sketch the core concepts, organize hierarchies, and discuss modeling decisions with your team.

Recommended tool: Chowlk

#### 2. Generate code from a diagram

If you have a chowlk diagram...

Drag and drop your diagram or click to choose your file



- UML based notation for OWL
- Converter
- Templates for draw.io

<https://chowlk.linkeddata.es/>

### Why?

The diagrams generated during this activity could be considered the first ontology version or the precursor of the ontology code, that is, this information could be considered provenance information for the final ontology and an important resource to understand the final ontology model.

### Why?

The diagrams generated during this activity could be considered the first ontology version or the precursor of the ontology code, that is, this information could be considered provenance information for the final ontology and an important resource to understand the final ontology model.

### Resources

- Ontology conceptualization, for example diagrams, concept maps, description based on logic languages (prior to the ontology implementation language code)

### Roles

- Ontology development team
- Domain experts
- Ontology user

## PRINCIPLES

Findable

Accessible

Interoperable

Reusable

#### Findable

FBD-Conc1: Diagrams, concept maps or other conceptualization resources should follow broadly adopted notations (e.g. Chowlk notation, UML, Grafoo, etc) or formats.  
I2Q6: Does the ontology provide information about the relation to or influence of other FAIR vocabularies?

P-Rec14: Standard vocabularies should be used to describe semantic artefacts.  
FBD-Conc2: In case potential ontologies to be reused are already known in this stage, include the ontology or ontology elements URIs in the ontology conceptualization diagrams.

#### Accessible

#### Interoperable

#### Reusable

FBD-Conc3: Metadata about diagrams, concept maps or other conceptualization resources should be provided.  
FBD-Conc4: Diagrams, concept maps or other conceptualization resources should be identified as provenance information from the ontology.  
G&P-6: Generate ontology metadata.  
G&P-8: Generate diagrams.  
FVF-11: Vocabularies meet domain relevant community standards.  
G&P-6: Generate ontology metadata.



## Ontology Encoding

During this activity, the ontology development team generates computable models in a machine readable format, for example the OWL language.

### Steps

#### 1. Design your ontology name

- Define your ontology URI, for example
  - Hash vs slash
  - Opaque URIs?

https://example.com/ontology#

How will the URIs of your terms be?

Opaque (example: `080_0001`)  Non-opaque (readable, example: `foaf:Person`)

I don't know

**Opaque URIs**

- Pro: better for compatibility with legacy systems and for multi-linguality.
- Con: difficult to read.

**Namespace URI check**

- Namespaces ends with #.
- The host `example.com` is not in the usual list of persistent providers (e.g. `w3id.org`, `doi.org`, `purl.org`, `data.europa.eu`, ...), but that does not mean it is wrong.
- Namespace length is 29 characters (< 40 recommended).
- Checklist: mark whether you will use opaque or readable URIs.

Declare your prefix

ssn

#### 2. Think about versions!

According to your namespace, here you have some options to define `versionIRI` (or similar)

Choose a versioning scheme:

Semantic Versioning (SemVer)

Version string or suffix:

1.0.0

Example: `https://example.com/ontology/1.0.0`

**Resulting version URI:**

`https://example.com/ontology/1.0.0`

Updated in metadata!

## Machine readable metadata assistant

### 3. Add basic metadata

Provide basic metadata and we will generate a Turtle snippet for your ontology.

Title:  Creator(s):

Description:

License:

Use the dropdown and start typing to search a license.

Generated metadata (Turtle)

```
@prefix ex: <https://example.com/ontology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix vann: <http://purl.org/vocab/vann/> .

ex: a owl:Ontology ;
dcterms:title "add ontology title" ;
dcterms:description "add ontology description" ;
dcterms:license "TODO: add license (e.g. CC-BY-4.0)" ;
dcterms:creator "add creator name" ;
owl:versionIRI <https://example.com/ontology/1.0.0> ;
vann:preferredNamespaceUri <https://example.com/ontology#> .
```

### 4. Generate code in different serializations

If you have a chowik diagram...

## Check whether the prefix is already being used



## Versioning schemes suggested

### Why?

During this activity the actual machine-readable version of the ontology is produced. It is key to apply FAIR Principles to this resource as it represents the main product of the ontology development process.

### Resources

- Ontology code is an formal implementation language, for example OWL or RDF(S)

### Roles

- Ontology development team

### PRINCIPLES

- Findable
- Accessible
- Interoperable
- Reusable

**Findable**

FVP-1: Vocabulary and their terms are assigned globally unique and persistent identifiers.

GAP-1: Design ontology name and prefix.

GAP-5: Use of permanent URIs.

Does the ontology have a "formal" identifier, i.e. a globally unique and potentially permanent identifier assigned by the developer (or developing organization)?

P-Rec1: Globally Unique, Persistent and Resolvable Identifiers must be used for Semantics Artifacts, their content (terms/concepts/classes and relations) and their versions.

Rule-6: Assign a unique and persistent identifier to (a) the vocabulary and (b) each term in the vocabulary.

FVP-2: Vocabularies and their terms have rich metadata.

GAP-6: Generate ontology metadata.

P-Rec3: A common minimum metadata schema must be used to describe semantic artifacts and their content.

Rule-7: Add vocabulary metadata.

GAP-6: Generate ontology metadata.

FQ01: Are the ontology metadata included and maintained in the ontology file?

FQ02: If not, are the ontology metadata described in an external file?

FQ03: Does that external file explicitly link to the ontology and vice-versa?

**Accessible**

FVP-6: Vocabularies and their terms are persistent over time and are appropriately versioned.

GAP-1: Design ontology name and prefix.

GAP-2: Decide between hash or slash URIs.

GAP-3: Decide whether to use opaque URIs.

GAP-4: Define an ontology versioning strategy.

GAP-9: Provide the ontology online in multiple formats (HTML, and ontology serializations).

P-Rec8: Human and machine-readable persistence policies for semantic artifacts metadata and data must be defined.

**Interoperable**

FVP-7: Vocabularies and their terms use a formal, accessible and broadly applicable, and preferably machine-understandable language for knowledge representation.

IQ01: What is the representation language used for the ontology and ontology metadata?

IQ02: Is the representation language used in a W3C Recommendation?

IQ03: Is the syntax of the ontology informed?

IQ04: Is the formality level of the ontology informed?

IQ05: Is the availability of other syntactic formats informed?

P-Rec9: Semantic artifacts must be made available as a minimum portfolio of common serialization formats.

P-Rec10: Foundational Ontologies may be used to align semantic artifacts.

P-Rec11: A standardised language should be used for describing high expressivity semantic artifacts.

P-Rec12: Semantic mappings between the different elements of semantic artifacts should be serialised in machine-readable formats.

Rule-8: Create machine readable representations of the vocabulary terms.

P-Rec10: Foundational Ontologies may be used to align semantic artifacts.

P-Rec14: Standard vocabularies should be used to describe semantic artifacts.

FVP-8: Vocabularies and terms use qualified references to other vocabularies.

IQ01: Does the ontology provide qualified cross-references to external resources/databases?

IQ02: If yes, are those cross-references well represented and to unambiguous entities?

IQ03: Does the ontology use valid URIs to encode some metadata values?

P-Rec10: Foundational Ontologies may be used to align semantic artifacts.

P-Rec12: Semantic mappings between the different elements of semantic artifacts should be serialised in machine-readable formats.

**Reusable**

FVP-2: Vocabularies and their terms have rich metadata.

FVP-9: Vocabularies and terms are described with a plurality of accurate and relevant attributes.

GAP-6: Generate ontology metadata.

RIQ5: How many ontology objects are defined using a property restriction or an equivalent class?

RIQ6: How much ontology objects provide provenance information with annotation properties (e.g. `author`, `date`)?

GAP-6: Generate ontology metadata.

P-Rec3: A common minimum metadata schema must be used to describe semantic artifacts and their content.

P-Rec17: Provenance must be clear for both humans and machines.

Rule-7: Add vocabulary metadata.

P-Rec3: A common minimum metadata schema must be used to describe semantic artifacts and their content.

P-Rec12: Semantic mappings between the different elements of semantic artifacts should be serialised in machine-readable formats.

FAIRbyDesign | GITHUB | HOME

Click on the ontology development phases or activities to get actionable steps to accomplish them

**Documentation**

During this activity the ontology development team in collaboration with the domain experts generates the human oriented ontology documentation (including metadata, descriptions, examples, etc.).

**Steps** Run all tests

- 1. Ontology documentation: all terms have labels**  
Enter ontology URI Submit Top Why? Principles
- 2. Ontology documentation: all terms have definitions**  
Enter ontology URI Submit



FAIRbyDesign | GITHUB | HOME

Click on the ontology development phases or activities to get actionable steps to accomplish them

**Online publication**

During this activity the ontology is made available online and accessible via its namespace URI as a machine-readable file and a human readable documentation using content negotiation.

**Steps** Run all tests

- 1. Ontology uses an open protocol (HTTP/HTTPS)**  
Enter ontology URI Submit Top Why? Principles
- 2. Ontology is available in RDF**  
Enter ontology URI Submit
- 3. Ontology has content negotiation for RDF**  
Enter ontology URI Submit
- 4. Ontology has HTML documentation**  
Enter ontology URI Submit
- 5. Ontology URI is resolvable**  
Enter ontology URI Submit
- 6. Consistent ontology IDs are employed**  
Enter ontology URI Submit
- 7. A version IRI is declared in the ontology metadata**  
Enter ontology URI Submit
- 8. Ontology version IRI resolves**  
Enter ontology URI Submit
- 9. Ontology found in community registry**  
Enter ontology URI Submit
- 10. Ontology metadata are accessible, even when the ontology is not**  
Enter ontology URI Submit

- **FAIR by Design framework**
  - *Methodology aligned with tools and existing recommendations*
  - *Actionable framework*
  - *However, increasing the FAIRness level of an ontology should be addressed along its **whole development life-cycle** (e.g., carefully documenting design decisions and requirements, including ontology metadata in the conceptualization phase, etc.)*

- **SOEL: Supporting Ontology Engineering with Large Language Models**
  - *How can LLMs support (FAIR) Ontology engineering*
    - *“Large Language Models for Ontology Engineering: A Systematic Literature Review”. Jiayi Li, Ziyuan Wang, Daniel Garijo, and María Poveda-Villalón. (Submitted to Semantic Web journal <https://www.semantic-web-journal.net/content/large-language-models-ontology-engineering-systematic-literature-review-1> )*
    - *34 papers || 49 experiments reviewed*
  - **Tooling** to facilitate metadata **updates**, keep **provenance** and **maintenance**: LLMs?
  - **Better orchestration** of services: **Agents**?
    - *“A Multi-Agent System for Explainable Ontology Generation. Jiayi Li, Ziyuan Wang, Daniel Garijo, and María Poveda-Villalón. (Submitted to LLM4KG OE)*



# Towards a FAIR-by-design ontology development framework

**María Poveda-Villalón**

Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

✉ [mpoveda@fi.upm.es](mailto:mpoveda@fi.upm.es)

📅 23<sup>rd</sup> March 2026

📍 AIOTI2026



This work was supported by the grant “SOEL: Supporting Ontology Engineering with Large Language Models” PID2023-152703NA-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF/UE”.