Report D2.9

"TLO/MLO Guidelines and Recommendations"

Grant Agreement: 958371

OntoCommons - Ontology-driven data documentation for Industry Commons, has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 958371.

Project Title	Ontology-driven data documentation for Industry Commons
Project Acronym	OntoCommons
Project Number	958371
Type of project	CSA - Coordination and support action
Topics	DT-NMBP-39-2020 - Towards Standardised Documentation of Data through taxonomies and ontologies (CSA)
Starting date of Project	01 November 2020
Duration of the project	36 months
Website	www.ontocommons.eu

D2.9 - TRO/MLO Guidelines and Recommendations

Work Package	WP2 | Top Reference Ontology
Task	T2.6 | Recommendations and Guidelines for TLOs and MLOs
Lead authors	Arkopaul Sarkar (ENIT); Claudio Masolo (CNR); Francesco Antonio Zaccarini (UNIBO)
Contributors	Francesco Compagno (CNR); Emanuele Ghedini (UNIBO); Stefano Borgo (CNR) and all WP2 Partners
Peer reviewers	Hedi Karray (ENIT)
Version	Final
Date	31/10/2023

Versioning History

Revision	Date	Editors	Comments
0.1	10/09/2023	Francesco Antonio Zaccarini	Creation of report structure; first report version.
0.2	11/10/2023	Claudio Masolo	Inclusion of material relative to the TRO framework
0.3	14/10/2023	Francesco Antonio Zaccarini	Inclusion of material relative to the formal framework and Bridge Concepts
0.4	17/10/2023	Arkopaul Sarkar	Inclusion of material relative to the Technical Principles and the formal framework
0.5	26/10/2023	Emanuele Ghedini & Stefano Borgo	Revision & final version.
0.6	31/10/2023	Hedi Karray	Technical approval
Final	31/10/2023	Nadja Adamovic	Final approval and submission

Glossary of terms

Item	Description
TLO	Top Level Ontology. A top-level ontology is an ontology (in the sense used in information science) which consists of very general concepts (e.g., the concepts of object, property, relation) that are common across all domains.
MLO	Middle Level Ontology. Mid-level ontologies are primarily intended to extend and specialise the concepts of TLOs towards a set of specific disciplines with the aim of providing a core shared vocabulary for lower-level modules. A MLO will generally provide a higher level of detail than a TLO, extending the taxonomical structure of the ontology more along on the horizontal dimension (i.e., sibling classes under the same superclass).

DLO	Domain Level Ontology. Domain-level ontologies are further specialisations of MLOs, even closer to the application level. The vast majority of their concepts is related to a specific discipline/domain, with a few instrumental/pragmatic exceptions, and vertical connectors.
ALO	Application-level ontologies are further specialisations of DLOs, explicitly or implicitly hinged on a specific set of application cases. They usually include concepts related to a specific set of intended tasks rather than concepts related to a discipline per se.
TRO	Top Reference Ontology. The top-reference-ontology is the upmost layer of the OCES, made up of a plurality of TLOs (i.e., BFO, DOLCE, EMMO) and the semantic alignment among them provided by T2.4 (the Meta-Ontology).
BC	Bridge-Concept. One of the core tools developed by the Consortium. A Bridge-Concept is a stand-alone ontological entity cum template which acts as a mediator and a point of reference both for (vertical and horizontal) semantic alignments and with respect to widely employed domain knowledge resources such as standards (given a double Hub-and-Spoke structure).
OCES (formal framework)	OntoCommons EcoSystem. The network of interconnected ontologies currently being developed by OntoCommons Consortium. It includes a core of selected aligned TLOs (i.e., BFO, DOLCE, EMMO), MLOs and DLOs. In the course of the project further ontologies will be produced, according to specifications outlined by the Consortium, to achieve full coverage of the NMBP-39 area.
SKOS	Simple Knowledge Organization System. A set of specifications and standards to express the relationships and concepts within knowledge organization systems, facilitating representation and management.
RDF	Resource Description Framework. A pervasive framework for modeling and describing resources on the web, allowing data to be exchanged among different systems in a machine-readable format. RDF uses triples, i.e., subject-predicate-object relationships, to represent data in a structured way.
RDF(s)	Resource Description Framework Schema. An extension of RDF that provides a basic vocabulary for defining the structure and semantics of resources. RDFS introduces vocabulary for defining classes, properties, and relationships between classes and properties.
OWL	Web Ontology Language. A semantic web language used for creating and managing ontologies. It is a key component in building the Semantic Web and supporting applications that require deeper understanding and interpretation of web-based information.

DL	Description Logic. A family of formal knowledge representation languages characterized by relatively high expressiveness, maintaining tractability (and decidability). OWL 2 DL is particularly relevant in this context.
SWRL	Semantic Web Rule Language. A rule-based language used in the field of semantic web and knowledge representation. It extends the capabilities of OWL by allowing users to create rules for drawing conclusions from RDF data.
FOL	First Order Logic. A formal mathematical language that allows for the representation of complex relationships (compared to propositional logic) and quantification over objects.
SPARQL	SPARQL Protocol and RDF Query Language. It is a query language used to retrieve and manipulate data stored in RDF (Resource Description Framework) format. It is designed for the Semantic Web (semantic data).
URI	Uniform Resource Identifier. A string of characters that provides a unique and standardized way to identify and access resources on the internet
IRI	Internationalized Resource Identifier. A specific type of identifier that can be used to uniquely name resources, concepts, or entities on the web. The related internet protocol standard builds on the Uniform Resource Identifier (URI) protocol, greatly expanding the set of permitted characters.
API	Application Program Interface. A set of rules, protocols, and tools that allows different software applications to communicate with each other.
FAIR **(Principles)**	Findable, Accessible, Interoperable, and Reusable. Guidelines and best practices for managing and sharing data and digital resources, in order to make data more discoverable, accessible, and usable. FAIR principles emphasize the importance of metadata, persistent identifiers, open access, and standardized formats.
OCEANS	OntoCommons Collaborative Editing and Authoring Knowledge Graphs. A platform developed by the OntoCommons Consortium used to record the knowledge generated in the project in a machine-readable format and visualized through knowledge graphs. The OCEANS platform is built on top of an open-source collaborative environment supporting RDF/OWL known as WebProtégé, which has been customized to better suit the needs of OntoCommons while still retaining most of its features.
LOT	Linked Open Terms.

	A methodology describing the activities that should be performed in the ontology development phases. This methodology typically follows Linked Data principles, making the terms machine-readable and linked to other relevant data sources, enabling easier data integration and sharing.
LOT4Ontocommons (LOT4OCES)	The version of LOT tailored for the OCES, integrating tools and methodologies developed by the Consortium.
NMBP	Nanotechnologies, Advanced Materials, Biotechnology, and Advanced Manufacturing and Processing. A collective term used to describe a broad spectrum of innovative technologies and scientific disciplines that encompass nanotechnology, advanced materials, biotechnology, and advanced manufacturing and processing. These areas often overlap and contribute to advancements in various industries and scientific research.
CQ	Competency Question. A concise and structured inquiry apt to evaluate a specific competence or knowledge in a particular area or domain. They can be used to define sets of requirements.
Ontology Artefact	An OWL ontology file containing terms or mapping axioms.

Keywords

OCES Framework, User Manual, Guidelines, Maintenance & Expansion, Technical Principles

Disclaimer

OntoCommons (958371) is a Coordination & Support Action funded by the European Commission under the Research and Innovation Framework Programme, Horizon 2020 (H2020). This document contains information on researched by OntoCommons Beneficiaries. Any reference to content in this document should clearly indicate the authors, source, organisation, and publication date. The document has been produced with the funding of the European Commission. The content of this publication is the sole responsibility of the OntoCommons Consortium, and it cannot be considered to reflect the views of the European Commission. The authors of this document have taken any available measure for its content to be accurate, consistent, and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur because of using its content.

Executive Summary

The objective of D2.9 is to provide best practices and guidelines for the utilization, exploitation, maintenance, and further development of the tools, methodologies, and infrastructure resulting from the efforts of WP2. This guidance is contextualized within the scope of the overall project. This white paper offers an overview of relevant practical and theoretical aspects related to (1) the TRO and (2) the MRO/Bridge-Concepts, presented in D2.7 and D2.8, respectively. As core contributions, D2.9 also includes a detailed discussion of (3) the OCES Framework (OWL), and (4) a list of technical principles endorsed by OntoCommons. This list serves as both effective recommendations that consider the current landscape and state-of-the-art literature and as requirements for stakeholders.

Table of Contents

List of Figures

1. Introduction

D2.9 can be considered a **summary of WP2's efforts** throughout the OntoCommons Project, including collaborations with other WPs and external experts. Its production is the primary objective of T2.6, and it represents a key document for the successful accomplishment of **O2.1** and **O2.4**. According to the DoA, D2.9 takes the form of a white paper and serves as both a **user manual** and a **manual for maintenance and development**. It provides best practices and guidelines for the **utilization, exploitation, maintenance, and further development of the tools, methodologies, and infrastructure** resulting from WP2's efforts.

Understanding the **guiding principles**, theoretical assumptions, and practical considerations that directed WP2's efforts has been deemed pivotal in achieving the aforementioned goals. Therefore, relevant elements have been included in the discussion. Nevertheless, D2.9 primarily focuses on **practical aspects**, offering clear workflows to achieve specific desired outcomes. As such, this manual caters to various levels of engagement for stakeholders, depending on their needs.

Consistent with the stated aims and to provide stakeholders with a **single point of reference** for accessing documentation related to the **OCES formal framework**, D2.9 offers a summary of results achieved in the context of T2.4 and T2.5, while elaborating on key points. Specifically, it provides an overview of aspects related to the TRO and the MRO/Bridge-Concepts, explored in D2.7 and D2.5/D2.8, respectively. Notably, this deliverable presents the first systematic aggregation of entry material on **Bridge Concepts**.

However, the main focus of D2.9 is the OCES formal framework. Its core, original contributions include a detailed discussion of the OCES Framework (OWL) and a list of **technical principles** endorsed by OntoCommons. The latter serves as both **effective recommendations**, taking into account the current landscape and state-of-the-art literature, and as **requirements for stakeholders**.

This is reflected in the organization of the document:

Sections 2 and 3 cover previous results: §2 is dedicated to the TRO (FOL to OWL) and §3 to the MRO, focusing heavily on Bridge Concepts. Specifically, §§2.1.x provide an introduction to the state of the art, contextualizing methodological choices. §§2.2.x and §§2.3.x present the workflow for the FOL alignments, and how they can be exploited to ground OWL alignments, respectively. Scientific details are reported for ease of reproduction of the results obtained. §2.4 offers guidelines for the expansion and maintenance of the TRO, and §2.5 relevant bibliographical references. §§3.1.x offer a general introduction to the MRO and Bridge concepts, while §§3.2.x discuss Bridge Concept engineering in detail, covering the selection of candidate terms, the informal characterization of Bridge Concepts, and their alignments. Guidelines for the maintenance of the MRO are also provided.

In Sections 4 and 5 the more original contributions are presented: §4 is dedicated to the OCES formal framework (focusing on the OWL implementation) and §5 to the Technical Principles endorsed by OntoCommons. Specifically, §§4.1.x provide an introduction to the OCES formal framework, focusing on the OWL environment and touching on aspects related to the subsumption of databases and knowledge graphs in the EcoSystem. §§4.2.x offer a manual for the use and maintenance of the framework, integrated with the LOT4OCES methodology. The sections also discuss points related to imports and inconsistencies management. §§5.1.x and §§5.2.x present the technical principles, recommendations, and requirements concerning IRI and Metadata, respectively.

Finally, the Conclusions (§6) summarize the contents of the document, offering brief parting comments and suggestions for the sake of completeness. Designed as a manual, the document is organized to allow stakeholders to read selected sections according to their specific needs. However, it is worth noting that the document's structure aims to facilitate the understanding of the rationale underlying the choices made. Readers may encounter difficulties if they attempt to go beyond the provided workflows without grasping the relevant concepts.

2. TRO

OntoCommons explicitly commits to a **pluralist approach** for the ontological representation of the domain of interest, meaning that more than one ontology for the same domain may be taken into account. It is difficult to avoid such heterogeneity in open and distributed systems because different actors may have different interests and points of view or they can analyse the domain at different levels of detail or granularity. The pluralist approach is **inclusive** with respect to users and enables **reuse of already available resources** but, to guarantee the systems' **interoperability** allowing for data sharing and harmonisation, it requires the different **models of the domain to be interlinked or at least partially integrated**.

To address this problem, OntoCommons relies on the **Ontology Commons EcoSystem (OCES)**, roughly an **integrated network of ontologies representing the domain at different levels of details**: top-level ontologies (**TLOs**) containing very general concepts and relations; domain-level ontologies (**DLOs**) addressing needs of specific demonstrators; and middle-levels ontologies (**MLOs**) smoothing the connection between TLOs and DLOs by introducing notions commonly used in clusters of DLOs. The general assumption is that a DLO refers and uses notions coming from a MLO that in its turn refers and uses notions coming from a TLO.

In this perspective, alignments and mappings can be introduced at all the levels. The (partial) **integration of high-level ontologies** (especially TLOs, but also MLOs) is however **particularly important** because the alignments concerning high-level notions (*i*) are central to clarify the considered general frameworks **preventing misunderstanding** also caused by the fact that TLOs (and sometimes MLOs) use quite **technical and/or philosophical oriented terminologies**; and (*ii*) they can be used to **indirectly map MLOs or DLOs through the mappings among the TLOs and/or MLOs they refer to**; and (*iii*) they can be used to **validate some direct mappings between MLOs or DLOs** that are already available.

2.1 Introduction and State of the Art

As discussed in the previous section, the integration of TLOs is central for the whole OCES. To approach such integration, OntoCommons relies on the Top Reference Ontology (**TRO**), i.e., **a set of TLOs together with a set of (partial) alignments/mappings between them**. More specifically, in the current version of the TRO, OntoCommons includes **BFO, DOLCE, and EMMO**. The TRO is however **open** to future extensions where other TLOs satisfying given criteria are included (see §2.6). For BFO, DOLCE and EMMO both a FOL and an OWL version are available, however no formal mappings between these TLOs exist (some papers discuss possible mappings between BFO and DOLCE but they are not formally grounded and they often concern only the main categories). A big effort in the WP2 goes then in the direction of **establishing such formal mappings**.

2.1.1 On Ontology Alignment and semi-automatic approaches

There exists a huge **literature on ontology alignment and matching**, (Euzenat and Shvaiko, 2013) and (Ehrig, 2007) are good systematic reviews. Besides clarifying and formally characterising what are the matching and alignment problems, such literature usually focuses on developing **algorithms for (semi-)automatic matching/alignment**. Unfortunately these **algorithms are not really helpful** for building the TRO.[1] Some details on these algorithms and on the nature of TLOs are necessary to understand why.

Algorithms can be roughly classified on the basis of the techniques they are based on. There are **two main classes of techniques**: *element-level techniques* "consider ontology entities [like concepts, relations, etc.] or their instances in isolation from their relations with other entities or their instances" (Euzenat and Shvaiko, 2013, p.79) while *structure-level techniques* "consider the ontology entities or their instances to compare their relations with other entities or their instances" (Euzenat and Shvaiko, 2013, p.81).

Structure-level techniques seem more promising in the case of TLOs which are highly structured ontologies, i.e., their classes are interlinked by means of a dense network of relations. Among structure-level techniques, instance-based techniques—which decide the matching of classes on the basis of their instances—are not very useful (*i*) because TLOs are not usually provided with instances and (*ii*) because some of their classes are not distinguished in terms of their extensions (i.e., classes often have an intensional nature). Among structure-level techniques, *graph-based techniques*, *taxonomy-based techniques*, and *model-based techniques* are in principle more adequate for TLOs because they take explicitly into account the structure of classes.

However, for reasons of computational efficiency, these techniques introduce some **limitations on the expressive power of the languages allowed to encode the ontology and (complex) mappings**.

[1] Likewise, there are prima facie reasons to suppose that there algorithms might not be effective even when MLOs are concerned. However, this should ascertained experimentally, and, at the present time, there is a lack of conclusive evidence.

Furthermore, the initial correspondences between concepts and relations in the ontologies are usually grounded on **terminological considerations**, i.e., they strongly rely on similarities between the natural language labels assigned to the concepts and the relations in the ontologies.

2.1.2 On the reasons to Prefer Manual Alignments

Both **these assumptions are problematic for the TLOs**. First, the classes (and the relations) used in TLOs are very general and abstract ant the used terminology used is often quite idiosyncratic and heterogeneous because it reflects specific philosophical or scientific theories. Furthermore, TLOs contains few dozens of classes and relations. **Inaccurate initial correspondences between them are then difficult to be corrected on the basis of structural considerations.** In addition, as shown in the deliverable D2.7, **TLOs can adopt very different classes and relations that require complex mappings**—and not *subclass* or *equivalence* mappings between classes as most of these automatic techniques refer to—**grounded on ontological, not terminological, considerations**. Second, even though BFO, DOLCE, and EMMO all have an **OWL version** that can be given as input to these algorithms, such versions **highly simplify the FOL ones** that contain a much richer characterisation of the adopted notions. In addition, as clearly make evident by the results in the deliverable D2.7, **the mappings between TLOs, being usually quite complex, require a language with a good expressive power to be encoded**.

If the TRO, besides its applicative role, is a **conceptual and ontological resource**—representing a starting point for **building new ontologies** (founding re-using and modularisation), a reference point for **rigorous comparison among ontological choices** (founding semantic integration and interoperability) and a **common framework for developing, analysing, harmonising, and integrating existing ontologies and metadata standard** (founding trust and community)—then the original richness of TLOs and mappings must be preserved as much as possible. This does not mean that the OWL versions cannot be integrated, but this integration, as we will discuss, must be grounded on the alignments done at the level of the rich FOL versions of TLOs.

2.2 FOL Alignments

The considerations in the previous section pushed us to follow in OntoCommons a **manual** strategy to build the TRO starting from the FOL versions of BFO, DOLCE, and EMMO and considering mappings that try to **match or maximally approximate the ontological commitments of the TLOs**.

2.2.1 Alignment Methodology: Workflow

In order to manually align the FOL versions of the TLOs in the TRO, taking inspiration from the methodology proposed in (Euzenat and Le Duc 2012) and refining a first proposal in the deliverable D2.4, we followed a **general methodology** that can be detailed in the following ten steps:

1. collect the FOL and OWL versions of two TLOs to be part of the TRO together with their explanatory documentation;
2. analyse the collected theories and the available documentation to ensure proper understanding of the intended interpretations of the primitive notions (categories and relations);
3. find and consider, if available, already existing formal or informal mappings between the two ontologies;
4. introduce formal mappings from one FOL ontology to the other (and vice versa);
5. evaluate the mappings;
6. improve the mappings by reiterating steps (4) and (5);
7. document and share the satisfying mappings;
8. use the mappings between FOL-versions to individuate the mappings between OWL-versions;
9. document and share the established OWL-mappings;
10. iterate steps (1)-(9) for all the couples of ontologies to be considered in the TRO.

For BFO and DOLCE we collected the versions in **Common Logic** (CL). The choice to use Common Logic is the result of several considerations including the fact that Common Logic is a consolidated standard (https://www.iso.org/standard/66249.html) and that the visibility and stability of standards increases users' trust in the OCES system. Furthermore the CL versions of BFO and DOLCE has been included, respectively, in the Part 2 and Part 3 of the standard "ISO/IEC 21838 Information technology – Top-level ontologies". This should assure that the versions of the ontologies included in the TRO are quite stable because, in principle, every change in a TLO in the TRO requires a re-evaluation of the mappings, a very costly activity.

2.2.2 Finding Correspondences among Notions

The **analysis of the documentation** and the examples in the ontologies helps in eliciting **informal correspondences** across the notions of the two ontologies providing a **basis for the formal mappings**. For instance, occurrents are described and used, in the BFO formal system and accompanying documentation, in a way that is conceptually similar to that of perdurants in the DOLCE formal system and documentation. This similarity is taken as an informal correspondence and leads to consider the potential alignment of the BFO occurrent category to the DOLCE perdurant category. (Note that such a correspondence might be suggested by common examples in the two ontologies, or even be based on the comparison of relevant relations which are, perhaps for different reasons, considered similar like the BFO relation continuantPartOf and the DOLCE relation temporary parthood).

2.2.3 Dealing with Inconsistencies in the Documentation

However, given the complex development process and the complex structure of TLOs, **incongruences across documents** relative to the same ontology can be found. Sometimes these incongruences can be managed by **relying on additional information** supporting what description in natural language is the most reliable of the authors' intentions. Sometimes there are also incongruences between the documentation and the axioms included in the formal theory. In these cases, given the formal nature of mappings and their evaluation (see below) we give **priority to the formal theory** (that can be tested for consistency).

2.2.4 Dealing with Lack of Detail on Aspects Relevant for the Alignment

Both the documentation and the formal theory (that usually only partially characterises the primitive notions of the ontology) might **not be detailed on aspects relevant for the alignment**. For instance, during the construction of the BFO-DOLCE alignment, the intended interpretation of some concepts was unclear to the alignment developers and **clarifying questions** were sent to the ontology authors to collect further information. This approach might not always be possible and, even so, it might not solve all cases (sometimes the authors of an ontology might not have a ready answer especially if the question raises foundational problems).

2.2.5 FOL Alignments: Scientific Addendum on steps 4 to 6

The conception of the formal mappings (introduced in the step (4) of the methodology) embraced in OntoCommons is based on two assumptions.

- **(A1)** The mappings aim to embed all (or most of) the individuals in the domain of quantification of the source ontology into the domain of quantification of the target ontology. The purpose is to **maximise the coverage of the entities** of one ontology by the domain of the other ontology. This is particularly important to make explicit the ontological commitments of the two ontologies and understand possible disagreements concerning the individuals accepted in the domain of quantification.
- **(A2)** Only **mappings that can be formalised in FOL as syntactic definitions** are considered. Mappings are then sets of syntactic definitions, i.e., FOL definitions of the primitive notions of the target ontology on the basis of the primitive notions of the source ontology. Meta-modeling techniques, like the application of abstraction processes or set-theoretical (second-order) constructions to enrich the domain of the source ontology with additional entities, are not considered. This assumption allows us to see mappings under the light of the standard logical notion of definitional extension, and to partially simplify and systematise the evaluation process in the step (5) of the methodology.

From a formal perspective, given two TLOs T1 and T2 in the TRO, the mapping from T1 (the source ontology) to T2 (the target ontology) extends T1 with a set of **syntactic definitions of the primitive notions** of T2. In some cases, not all the primitives of T2 can be defined in terms of the primitives of T1, i.e., T1 is not expressive enough to recover the whole T2. The alignment can then be partial because of this lack of expressive power of T1.

The evaluation of the mappings, the step (5) of the alignment methodology, is admittedly the most critical one. A formal way to evaluate the mappings (from the source ontology T1 to the target ontology T2) considered in OntoCommons consists in analysing how much of T2 is **preserved by the mappings**, i.e., to study which axioms of T2 hold or fail to hold in T1 once it is extended with the definitions of (part of) the primitives of T2. The fact that, even for the primitives for which an explicit syntactic definition exists, not all the axioms regulating them are preserved is a second source of the partiality of the mappings.

In this perspective, assumption (A1) is useful to **highlight genuine differences between the ontological commitments** of the two ontologies and which entities of the source ontology are problematic in, or even incompatible with, the target ontology. Assumption (A2) is relevant also from a practical perspective because it opens the possibility to use **theorem provers** to help to automatically check what axioms of the target ontology are preserved.

There is however an **intrinsic difficulty in having an evaluative quantitative approach**. The preservation of some axioms may be more important that the one of other axioms. One can then think to assign **weights** to axioms, however it is not clear to us how such weights can be established. More importantly, the **non-preservation of some axioms could not be due to the (bad) quality of the mappings but to a genuine ontological disagreement between the two theories** considered (see the analyses of the mappings in the deliverable D2.7). Vice versa, the **preservation of some axioms could be due to the introduction of mappings that work at the formal level but are not ontologically plausible or well founded**. Thus, the evaluation of the mappings seems a complex process with a **qualitative** nature that is not reducible to a purely formal dimension but requiring the understanding of the coherence of the mappings with respect to the intended interpretations of the primitive notions of the two ontologies.

2.3 FOL to OWL

2.3.1 Premises

In order to facilitate the exploitation of the FOL alignments via the variety of tools that support semantic web languages, the TRO can be used to create a framework for the TRO in OWL 2 (in short TRO-OWL). In this framework, the OWL 2 version of the TLOs are integrated with a set of OWL 2 mappings grounded on the FOL mappings, with each TLO and the set of mappings being integrated in a **modular fashion**.

The **reduced expressiveness** of OWL 2 DL does not allow to fully represent the conceptualization behind the FOL versions of the TLOs and their alignments; yet, by enforcing computability, it massively increases the usability of the ontologies. It is important to notice that the majority of links between the *primitive relations* adopted by different TLOs, a crucial aspect of the TRO, cannot be expressed in OWL 2 but can be at least **partially captured in SWRL**. Without sacrificing computability, the RL extension preserves some of the advantages provided by expressive formal systems; as such, the SWRL mappings can be exploited to exchange data on demand, or for local applications, if not as the standard. The mapping themselves are then divided into two parts: a first one, OWL 2 DL compliant, containing simple taxonomical relations, and second one, SWRL-based, including more complex connections concerning relations.

The whole OWL/SWRL framework is fully Protégé compatible and offers to the end user the possibility to navigate through the TLOs, their mappings and lower-level ontologies connected to them. The OWL mappings, derived from rigorous FOL alignments, ground partial links between the TLOs' respective taxonomic trees. The TRO-OWL may also constitute a development framework for semantic web ontologists to build and test OntoCommons compliant ontologies.

2.3.2 OWL Alignments: Workflow and Underlying Assumptions

Concerning the step (8) of the methodology—i.e., the individuation of the mappings between the OWL versions of the ontologies on the basis of the mappings between the FOL versions—we followed a formal approach retaining only **"safe" mappings**, i.e., mappings grounded on explicit theorems provable on the basis of the FOL mappings.

First of all, OWL 2 taxonomical mappings are mainly represented by means of subClassOf(T1:C1 T2:C2) statements where T1 (T2) is the OWL version of the source (target) TLO and C1 (C2) is a class in T1 (T2). Mappings are "safe" in the sense that a subClassOf(T1:C1 T2:C2) statement is introduced in TRO-OWL only when in the FOL version of the source TLO extended with the FOL mappings (including the FOL definition of C2) it is possible to prove that the class C1 is a subclass of C2 but it is not a subclass of any subclass (in the target ontology) of C2, i.e., C2 is the minimal class of the target ontology subsuming C1. Similarly in the case of equivalences.

Second, in the case of the mappings concerning relations, we empirically found that usually a mapping can be established **only when some restrictions on the arguments of the relations are in place**. To express these mappings one needs to move from OWL 2 DL to SWRL where it is possible to introduce rules with the following form (where T1 and T2 are the OWL versions of the source and target TLO, respectively, R1 is a relation in T1, R2 a relation of T2, C1 and C2 are classes in T1).

T1:R1(?x,?y), T1:C1(?x), T1:C2(?y) => T2:R2(?x,?y)

capturing the fact that R1 can be considered as a subrelation of R2 only when the first argument of R1 is restricted to C1 and the second argument of R1 is restricted to C2. As said, formally these SWRL mappings are less effective that OWL mappings but they can be exploited to exchange data on demand or for local applications. As for taxonomical OWL mappings, the SWRL rules are grounded on the FOL-mappings.

Third, the mappings between TLOs based on very different ontological commitments, e.g., the DOLCE-EMMO mappings (see the deliverable D2.7), usually require **complex constructions** which are based on the definition of new intermediate concepts and relations. In these cases, only fairly uninformative links can be established considering uniquely the original classes in the ontologies. The need for constructive extensions to bridge the gap between the two ontologies needs then to be taken into account also in the OWL-version of the mappings. A methodology we explored in OntoCommons relies on the introduction of new ***hidden classes*** that intuitively generalise/specify some classes already present in the ontologies or "simulate" negation (in the sense that they collect instances of a class C that does not instantiate a set of subclasses of C). These hidden classes are added to the OWL versions of the ontologies and are mainly characterized via subClassOf and Disjoint statements or by making use of SWRL rules.

The hidden classes and the statements and rules used to characterise them are not strictly part of the original ontologies, though they are operationally considered as belonging to be part of them. Their introduction requires then particular care, in order to avoid unintended changes to the original ontologies (it might also be possible to make use of rules involving classes and relations from a plurality of ontology to achieve more specific links, however, such an endeavour requires particular methodological care, and it was decided not to pursue an experimental line, without proper prior investigation, in the context of the project). Through the use of these hidden classes one can then partially capture mappings based on complex constructions relying on intermediate notions. For some examples one can refer to the Section 10.2 of the deliverable D2.7.

2.4 Update and Maintenance of the TRO: General Notes

The introduction of a new TLO T in the TRO requires to iterate all the steps in the methodology introduced in section 2.2 for all the couples of TLOs (T,Ti), where Ti is a TLO already included in the TRO. This is a complex and time consuming process. To mitigate the effort one may introduce direct **mappings only with a subclass of the TLOs** in the TRO and then **rely on the alignment already in the TRO** to find indirect mappings with the other ontologies. We followed this strategy for the BFO-EMMO alignment that can be obtained by composing BFO-DOLCE and DOLCE-EMMO mappings. This approach is however only partially satisfying because, given the fact that the mappings can be partial (with respect to the covering of primitives and the preservation of the axioms of the target ontology) **the composition of mappings deteriorate their quality and possibly hides some direct mappings** (not obtainable via the intermediate of a third TLO).

A TLO is considered as a candidate to be included in the TRO only when the following requirements are met:

1. both a FOL version and an OWL version are available together with good explanatory documentation;
2. it is stable (to avoid frequent maintenance of the mappings, an extremely time consuming process);
3. its development and exploitation are continuously supported by the developers which commit to collaborate to the establishment of the FOL and OWL mappings with at least another TLO already in the TRO;

4. it is adopted by a substantial number of MLOs and/or DLOs, i.e., it is a reference point at least in some applicative domains.

Another delicate aspect concerns the understanding of how the **updating of a TLO in the TRO impacts the mappings**. From a purely theoretical perspective, every change in the axioms of a TLOs modifies the intended interpretation of the primitives and then, in principle, a re-evaluation of the mappings concerning this TLO is required. This pertains not only the mappings with the other TLOs in the TRO but also the mappings with the MLOs/DLOs (referring to the updated TLO) in the OCES and the mappings with the bridge concepts. Regarding the mappings with the other TLOs, one can try to mitigate the effort by circumscribing the impact, e.g., the impact of a change in the source ontology can be evaluated by understanding what proofs of theorems in the source ontology extended by the mappings rely on the axioms that has been changed and verify if the affected theorems still hold or does not hold anymore. The impact of a change in the target ontology can require to prove new theorems, namely the ones relative to the new axioms (and discard old theorems, the ones relative to the deleted axioms). The mappings from the MLOs, DLOs, or bridge concepts towards the TLOs usually have a taxonomical nature. Even though one should always check the extend of the changes, such taxonomical links among categories are usually more robust and less affected by changes of axioms that do not radically modify the intended interpretation of the categories of the TLO.

More drastic changes could so deeply impact the intended interpretations of primitive notions to invalidate some syntactic definitions considered in the mappings between TLOs or invalidate taxonomical links with MLOs, DLOs, or bridge concepts. In these cases, one needs to individuate new syntactic definitions and/or new taxonomical links and then evaluate them. As such, the OntoCommons Consortium recommends TLO developers to **avoid making major overhauls** once their ontologies are part of the EcoSystem.

2.5 Bibliographical References for §§2.x

- (Ehrig, 2007) Ehrig, Marc. Ontology alignment: bridging the semantic gap. Vol. 4. Springer Science & Business Media, 2007.
- (Euzenat and Le Duc 2012) Euzenat, J., Le Duc, C.: Methodological guidelines for matching ontologies. In: Suárez Figueroa, M., Gómez Pérez, A., Motta, E., Gangemi, A. (eds.) Ontology Engineering in a Networked World, pp. 257–278. Springer, Heidelberg (2012). Chap. 12 (pp. 55, 350)
- (Euzenat and Shvaiko, 2013) Euzenat, Jérôme, and Pavel Shvaiko. Ontology matching (second edition). Vol. 18. Heidelberg: Springer, 2013.

3. MRO and Bridge Concepts

The aligned TLOs, i.e., the TRO, already establish a certain **degree of interoperability** among lower-level ontologies. However, it's essential to recognize that reasoning and data 'flow downwards' from the roots to the leaves, specifically from a class or property to its subclasses or subproperties.

Consider, for example, two ontologies, O' and O'', where the class C' 'Animal' from O' is linked to the class C'' 'Fauna' in O'' through a class equivalence relation, such as 'owl:equivalentClass.' Additionally, let's assume that the class C''' 'Cat' is a subclass of C' in O', and the class C'''' 'Feline' is a subclass of C'' in O''. Given the alignment between 'Fauna' and 'Animal,' it becomes possible to import individuals from C''' into C'', and individuals from C'''' into C'. Consequently, reasoning extends from the now equivalent C' and C'' to C''' and C''''.

However, an essential point to note is that no data or reasoning can flow from C''' to C'''' or vice versa since they are both positioned 'below' the pipeline. To help stakeholders with no prior knowledge of data management and alignment understand this concept, an analogy using liquid flow can be quite illustrative.

It's worth mentioning that alignments among superclasses can serve as **facilitators** for establishing alignments among leaf classes. This is one of the reasons cited in the literature for the adoption of TLOs in Ontology Hubs, and its effectiveness is extensively discussed in relevant literature. In addition to proposing alignments, TLOs, including the TRO in the OCES, can also serve as a means to validate proposals or, at the very least, identify problematic alignments that may lead to inconsistencies. Arguably, OntoCommons' pluralistic TRO offers **multiple checks**, making it more beneficial in this regard.

Nonetheless, establishing links among ontology entities within lower-level ontologies is crucial to achieve complete data interoperability and prevent the loss of valuable information, particularly in industrial contexts.

Regrettably, the strategy used to align TLOs is hardly replicable when it comes to Mid-Level Ontologies (MLOs) and is entirely unfeasible for Domain-Level Ontologies (DLOs) and Application-Level Ontologies (ALOs). This is due to challenges related to the availability of First-Order Logic (FOL) versions of the ontologies, as well as the fact that most DLOs are lightweight and often lack the necessary documentation to support alignments. Manual pairwise mappings are out of question, given the **substantial number of ontologies** involved (each including a great number of concepts) and the fact that the **number of required alignments increases exponentially**, whereas even a linear growth is arguably problematic. As it should be patent, this issue primarily pertains to **scalability**.

Standard automatic and semi-automatic approaches have the limits outlined in §2.1.2, and are presently too imprecise/offering connections with are **not informative enough** given the aims of the project. It ought to be taken into account, that the OCES aims to connect ontologies belonging to **different domains**, possibly with only marginal -yet meaningful (with respect to value-chains)-

overlaps. Adding onto that, OntoCommons aims to connect ontologies at **different levels**, whereas terminological approaches struggle to deal with that even if supported by external resources and glossaries, when technical jargon is concerned. As such terminology-heavy solutions are not recommendable.

In the context of the project, it was considered whether a mixed semi-automatic approach, using the TRO alignment to avoid inconsistencies and to add further information for the mappings, could be exploited to good results. However, it was not possible to test this methodology during the project, though plans were made to potentially investigate this in cooperation with the **Ontology Alignment Evaluation Initiative**. It should also be noted that there was an attempt to integrate and exploit automatic alignment tools (in an ancillary role) in the context of the endorsed methodology, as discussed in §3.3.1.x.

The OntoCommons Consortium developed a **tool, Bridge Concepts**, to address the core issue related to scalability, as well as others (both OCES-specific and well-known in the literature) pertaining to **ontology-alignment and interoperability**, some of which are listed below:

- Terminologies' opaqueness (especially cross-domain).
- Concepts' indeterminacy
- (Pragmatic) Ontology entities' under-characterization (informal).
- (Pragmatic) Ontology entities' under-characterization (formal).
- Presence of Mistakes in Ontologies (especially light-weight).
- Lack or inadequateness of the relevant documentation.
- Domains' idiosyncrasies (ontology coverage etc.).
- Alignment Scenarios idiosyncracies (TLO-TLO vs TLO-DLO).

It is also worth noting that **new ontologies** were to be developed in the context of the OntoCommons Project to repair gaps in coverage; as such, a proper methodology had also to facilitate new ontologies' engineering, as well as the reuse of existing concepts, modules, and entire ontologies, as per the GA, and in line with the core OntoCommons Principles.

Initially developed to establish the MRO (playing the role of the alignments among TLOs), Bridge Concepts have been adopted (with some adaptations) across the board as the glue that connects the entirety of the OCES below the TRO.

See also D3.5 for other strategies taken into account by the Consortium.

3.1 Bridge Concepts: Core Roles

(The following discussion builds on work already presented in D2.5, D2.8, D3.6, D3.7, D3.8, as well as various white papers, webinars and presentations. The reader can also refer to the following paper: de Baas, A. et al (forthcoming), *Review and Alignment of Domain-Level Ontologies for Materials Science*, IEEE Access.)

Bridge Concepts find their foundational principles in **anti-representationalist perspectives** on language and **Carnapian conceptual engineering**. They operate on the premise that **everyday concepts/terms inherently lack determinate semantics**, and are characterized by **inherent fuzziness**; in fact, the position could also be described as endorsing the view that concepts/terms should only be considered as they are employed in standard practice in a given context, and they are inherently unstable even for single individuals.

This position reflects the perspective that communication is founded on **acts of negotiation**, rather than universally shared and precisely defined semantics, or well-anchored entities; linguistic practices themselves cannot be analyzed or reduced in any simple or straightforward manner. The primary objective of Bridge Concepts is to translate and adjust this perspective for a **machine-readable, rigid, environment**, accommodating a degree of negotiation and imprecision within pragmatic boundaries – which are defined by specific use-case scenarios.

The most effective way to approach Bridge Concepts is by focusing on the challenges they were tailored to address, focusing on their two core roles. These roles are succinctly outlined as follows:

- **Supporting Scalable Ontology Alignments**: Bridge Concepts operate as stand-alone ontology entities, crafted and characterized independently of any specific ontology framework, thereby establishing their ontology-neutral nature. They occupy a central position within a hub-and-spoke architecture, supporting robust semantic alignments across multiple ontologies. Functioning as **data pipelines** or **minimal ontology content patterns,** they are initially aligned TLOs through robust semantic alignments, and then to relevant lower level ontologies - with one or more concepts (ideally a subclass and a superclass, or, even better, an equivalent class). This strategic configuration significantly reduces the volume of requisite alignments, effectively addressing scalability concerns, a critical aspect given the ambitious scope of the project. Importantly, it's worth noting that these tools create **networks of connections** rather than engendering super-ontologies or super-standards. This approach obviates potential interoperability issues at the meta-level, ensuring a firm and conflict-free foundation for the project.
- **Acting as a User-Friendly Interface**: Bridge Concepts also double as practical dictionaries tailored for ontology implementation. Their role is to ensure accessibility for a diverse array of stakeholders and domain experts, offering reference points via connections to established **golden standards**. They are characterized in a way which **leverages domain experts' background knowledge**, with the aim of providing pragmatically unambiguous and simple ways to determine whether e.g., a given individual falls within the extension of a class or not. The focus is on potential **ambiguities which could compromise ontology use in practical use cases**, rather than on providing a unique, rigorous definition of an extension, connection to other concepts equally formally defined, which are generally limited by the escalating complexity required (the only exception being Bridge Concept clusters; more on this infra). Likewise, characterizations and alignments are extensively

commented: keeping track of the rationale motivating certain choices greatly **facilitates negotiation**, reducing the potential for human errors down the line.

Let us introduce the following distinction between formal and informal characterizations:

- **Formal Characterizations**: These encompass the hierarchical structuring of an ontology, including classes, relationships, and axiomatization. The emphasis is placed on mathematical and structural aspects.
- **Informal Characterizations**: In contrast, informal characterizations encompass elements such as natural language labels, descriptions, comments, and contextual interpretations. These elements are pivotal in enhancing human understanding and accessibility to the ontology.

Bridge Concepts primarily operate within the informal sphere (they lack formal characterizations upon creation), aiming to bridge the gap between ontologies characterized by differing formal descriptions of concepts that encompass the same individuals. This initiative is instrumental in harmonizing diverse ontologies within the project's framework.

The principles of **scalability and controllability** emerge as critical considerations for any approach seeking to establish a comparable network of ontologies. While automated and semi-automated tools may offer promise, especially given additional information provided by the TRO, it is essential to underscore that **alignments represent a substantial investment**. Therefore, precision remains a paramount concern, particularly in the realm of the Nanomaterials, Advanced Materials, and Biotechnology domain.

It is possible to further distinguish two **Bridge Concepts' subroles**: they are capable of connecting **Ontology Levels** (support **vertical** ontology alignments) as well as **Ontologies at the same level** (support **horizontal** ontology alignments). All Bridge Concepts are capable of playing **both of these roles at once** – this being one of their core strengths; however, it might be worth classifying them in the aforementioned way for practical purposes.

It is imperative to appreciate the unique roles played by vertical and horizontal connections. Vertical alignments predominantly facilitate the vertical (downward) flow of **reasoning**. In contrast, horizontal alignments foster **data exchange at critical junctures**, enabling seamless sharing of information across ontologies.

Vertical connections occupy a central position in realizing OntoCommons' overarching objective. This ambition entails **linking a vast array of concepts from lower-level ontologies through a concise set of Bridge Concepts**. This strategy hinges substantially on the existing intra-ontology semantic relationships, exploited by Bridge Concepts. As it will be discussed *infra*, the alignments with the TRO greatly enhance the methodology as a whole, allowing for better, more controlled, and thus precise, horizontal connections, and facilitating the establishment of links in general (in line with the discussion produced in §3: it is noteworthy that horizontal connections are inherently intertwined with their vertical counterparts within the proposed framework. The comprehensive **methodology built upon Bridge Concepts heavily relies on the TRO mappings,** exploiting the peculiarities of the OCES pluralist framework as well as the quality of the FOL alignments on which the TRO rests).

While this approach significantly reduces the requisite number of alignments for addressing scalability issues, it does entail potential **trade-offs** related to the loss of certain reasoning capabilities. Achieving equilibrium among entity coverage, the quantity of Bridge Concepts, and the exploitation of reasoning represents a complex undertaking.

Horizontal connections, conversely, assume a pivotal role in ensuring efficient data sharing. They are often perceived as the **bedrock of interoperability** by stakeholders. In addition to fostering enhanced interoperability, horizontal connections facilitate the retrieval of specialized reasoning and conceptual domains, thereby enriching same-level ontologies and the overarching OCES. Moreover, Bridge Concepts specializing in horizontal connections serve to bolster the effectiveness of reasoning derived from higher-level ontologies. To optimize horizontal connections, a judicious **balance** must be struck between the **quantity of required Bridge Concepts and the volume of shared data**. Identifying the most critical data and the mechanisms for its transmission assume critical significance in this context.

It is paramount to bear in mind that these two roles are not mutually exclusive. Bridge Concepts are inherently multifunctional, and the majority of them often serve both roles, albeit to varying degrees. The extent to which they fulfill these roles is contingent upon the chosen perspective (the focal ontology level) and the internal organizational structures of the involved ontologies. **Contextual and circumstantial parameters exert a significant influence on their operational dynamics**.

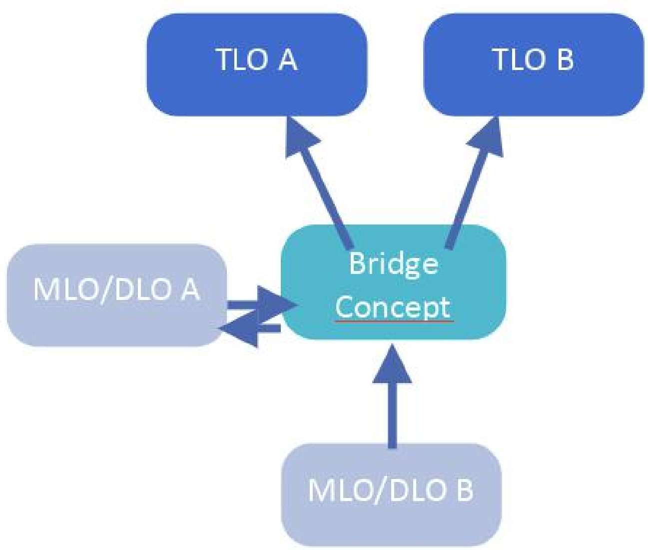

Figure 1 - Placement of Bridge Concept in OCES stack of ontologies

3.1.1 Bridge Concepts Template in Detail

Given their second overall role, it is pivotal for Bridge Concepts to be provided in a format which is **accessible** by non-ontologists. The following **template** acts as a **human-friendly interface** for Bridge Concepts, making the documentation overall more accessible to all stakeholders.

NEW CONCEPT NAME[1]

(use the preferred label, or IRI name, provided in the first table as title)

Concept elucidation

GENERAL CONCEPT INFO:

IRI:	*Suggested entity new IRI.*
OWL Type:	*Class / ObjectProperty / Individual.*
Concept Elucidation:	*Natural language definition of the concept (elucidation).* *Here the concept that we want to introduce is expressed as precisely as possible, making references to knowledge domain resources, including instance and usage examples when relevant.*
Labels:	*Labels used to address the concept, ordered as:* *i) preferred (one) (the label to primarily used to shortly refer to the concept)* *ii) alternative (multiple) (labels that are commonly used to address the concept in practice, even if they are used with narrower of wider sense)* *iii) deprecated (multiple) (labels that are misleading with respect to the concept, because of misuse, ambiguity or too wide meaning).*

Knowledge domain resources

KNOWLEDGE DOMAIN RESOURCES:

Related Domain Resources:	*Existing domain resources (e.g. standards, books, articles, dictionaries) that defines or are related to the concept (provide reference to the resource and quote the relevant informational content).* *More than one resource can be reported.* *These resources are aimed to support the choice of the above concept choice and elucidation.*
Comments:	*Explain the motivations behind the concept definition with reference to the domain resources, underlying similarities and differences.*

Definition by an existing formal ontology

ALIGNMENTS TO EXISTING ONTOLOGIES:

Target Ontology:	*Existing IRI of the ontology that will express the concept according to its logical framework (concept alignment).*
Related Ontology Entities:	*List of terms and IRIs of the Target Ontology entities that are relevant for the concept (documentation is supposed to be accessible through the target ontology).*
Mapping Elucidation:	*Natural language description of the mapping choice and motivations.*
Semantic Relation Level:	*The level of semantic relationship between the Concept and the Target Ontology entities:* - *Equivalence (strong mapping) (e.g. owl:equivalentClass, owl:equivalentProperty)* - *Strong Hierarchical (e.g. rdfs:subClassOf, rdfs:subPropertyOf)* - *Weak Hierarchical (e.g. skos:narrower, skos:broader)* - *Similarity (e.g. skos:related).*
Mapping Axioms:	*Proposed mapping axiom (or axioms) between the Concept entity and the Target Ontology entities in a OWL2 compliant syntax (e.g. Turtle, Manchester, RDF/XML, Functional-Style, OWL/XML).*

Figure 2 - Bridge Concepts' Template

Despite being **human-friendly**, the template is designed to be **easily transposed in machine-readable environments** (in the context of the project, it is stored both in markdown format and implemented as annotations in .owl given well defined procedures). It is divided into three main sections, **each targeting different stakeholders**: (1) General Information & Elucidation; (2) Related Domain Resources; (3) Alignments with concepts from existing ontologies. More details are reported in what follows:

1. **General Information**
    a. *Concept Name*

    The label, preferred label, or Internationalized Resource Identifier (IRI) title used to identify the bridge concept.

    b. *IRI*

    The proposed IRI for the bridge concept.

    c. *OWL Type*

    A value between Class, ObjectProperty, or Individual. In the course of the project, the focus was on Classes, for the sake of simplicity.

    d. *Concept Elucidation*

    This provides a natural language, informal definition of the concept, intended to be easily understood by domain experts. Elucidations should align with common knowledge and domain resources, avoiding references to other ontology entities (i.e., they should be ontology neutral). Ideally, they should also remain ontologically neutral (avoid commitments beyond the domain they pertain to) and concise, with the inclusion of diverse usage examples and the explicit addressing of potential ambiguities, focusing on cases relevant for (the expected) ontology usage.

    e. *Labels*

    Labels used to refer to the concept, categorized as follows: (i) preferred label – the primary label for referring to the concept, combining intuition and informativeness; (ii) alternative labels – multiple labels commonly used to address the concept, even if they have narrower or wider meanings; (iii) deprecated labels – labels that may be misleading or encourage misuse. Notably, hidden labels can be included to support queries.

2. **Knowledge Domain Resources**
    a. *Related Domain Resources*

    It lists existing domain resources, such as standards, books, articles, and dictionaries, considered during the development of the bridge concepts. The template includes static references to these resources and quotations of relevant content. Multiple resources can be reported, especially widely shared ones that may have influenced users. These resources not only guide the engineering phase but also help domain experts position the bridge concepts, serving as points of reference and enhancing conceptual clarity.

b. Comments

Comments in this section explain the motivations behind the concept elucidation, drawing from domain resources and highlighting similarities and differences.

3. **Alignment to Existing Ontologies**

a. Target Ontology

This is where the IRI of the ontologies that encompass ontology entities supporting the establishment of semantic connections with the bridge concept is listed.

b. Related Ontology Entities

A list of identifiers for specific ontology entities that the bridge concept is semantically connected to, promoting FAIR-ness.

c. Mapping Elucidation

This section provides a natural language discussion of the mapping choice and the underlying rationale. It includes information about alternative mappings considered and evidence gathered, facilitating third-party evaluation and validation of the proposed connection, and contributing to the clarification of the bridge concept.

d. Semantic Relationship Level

The strength of the semantic relationship with a specific ontology entity is detailed here. It can be: (a) Equivalence (strong mapping), e.g., owl:equivalentClass, owl:equivalentProperty; (b) Strong Hierarchical, e.g., rdfs:subClassOf, rdfs:subPropertyOf; (c) Weak Hierarchical, e.g., skos:narrower, skos:broader; (d) Similarity, e.g., skos:related. Ideally, for each ontology, a bridge concept should be connected to a set of broader concepts (usually one) and a set of narrower concepts (at least one). The strength of the relations and the informativeness of the mappings determine the quality of the overall alignment, reflecting how much data and reasoning can potentially be shared with other ontologies.

e. Mapping Axioms

Proposed mapping axiom(s) between the Concept entity and the Target Ontology entities are provided in an OWL2 compliant syntax, such as Turtle, Manchester, RDF/XML, Functional-Style, or OWL/XML. Note that bridge concepts support complex alignments that may involve properties.

Summing up, the template developed by the OntoCommons Consortium is intended to make them accessible, as well as to **facilitate the engineering and alignment** of bridge concepts (compare with §3.3.x). It is designed with ontology implementation in mind, and it adheres to the **FAIR**

principles. This approach can **address challenges related to the lack of documentation** in ontologies once mappings are established, thus going ways towards solving one of the core issue currently plaguing ontologies, and greatly impeding reuse and accessibility. The structure of the template is user-friendly, with the first part containing essential information for all users and the subsequent sections catering to domain experts and ontologists, respectively.

3.2 Engineering Bridge Concepts

What follows is an iterative workflow for the engineering, and subsequent alignment, of Bridge Concepts, for practical use:

PHASE I: Groundwork

1. Individuation of Candidate Bridge Terms:

   Begin by identifying candidate bridge terms through the analysis of a set of ontologies. Standards and other domain resources, or by focusing on the results of competency questions.

PHASE II: Circumscribing the Target

2. Initial Selection of a Target:
   Identify an appealing vaguely-defined concept that underlies the selected term and which is allegedly in line with the resources it was extracted from.
3. Preliminary Concept-Precisification Feasibility Check:
   Determine the feasibility of precisely defining the concept for ontology use.
   a. Success leads to Phase III.
   b. Failure allows you to either revisit step 2 or relinquish the candidate term.

PHASE III: Informal Elucidation of the Bridge Concept

4. Tentative Elucidation of the Concept:

Provide an initial elucidation of the concept, drawing inspiration from domain resources or potential alignments with concepts from a set of ontologies.

5. Definition Evaluation:
   a. Success leads to Phase IV.
   b. Failure results in definition bargaining, revisiting step 4, or relinquishing the concept.

PHASE IV: Refinement

6. Investigation Targeting Non-Ontological Golden Standards:

Investigate non-ontological golden standards based on the initial elucidation.

7. Investigation Targeting Related Concepts in Ontologies:

Investigate related concepts in ontologies based on the initial elucidation.

8. Domain Resources Evaluation and Selection:

Select domain resources for the relevant section of the template.

9. Elucidation Re-Evaluation:
    a. Success leads to step 10.
    b. Failure results in definition refinement to accommodate "new" domain resources, revisiting step 9, or returning to step 8 if the new domain resources pose challenges.
10. Elucidation-Comment with References:

Provide a comprehensive elucidation-comment with references to domain resources, with a focus on standards.

PHASE V: Alignment

11. Selection of the Alignment Targets:

Choose alignment targets, including the Top-Level Ontologies part of the TRO by default, and desired lower-level ontologies.

12. Tentative Vertical Alignments with the TRO:

Establish tentative vertical alignments with the TRO (links with specific concepts from each TLO)

13. Vertical Alignments Check:
    a. Success leads to step 14.
    b. Failure allows you to revisit step 12 or return to PHASE III if issues persist.
14. Vertical Alignments-Comment Proposal:

Propose comments for vertical alignments.

15. Vertical Alignments-Comment Evaluation:
    a. Success leads to step 16.
    b. Failure allows you to revisit step 14 or return to step 12 if necessary.
16. Tentative Horizontal Alignments with Selected Lower Level Ontologies:

Establish tentative horizontal alignments with the selected lower-level ontologies.

17. Horizontal Alignments Check, Considering Vertical Alignments Constraints:
    a. Success leads to step 18.
    b. Failure allows you to revisit step 16 or return to either step 12 or PHASE III if issues persist.
18. Horizontal Alignments-Comment Proposal:
    Propose comments for horizontal alignments.
19. Horizontal Alignments-Comment Evaluation:

   a. Success leads to Phase VI.
   b. Failure allows you to revisit step 20 or return to step 18 if issues persist.

PHASE VI: Completion

20. Final Check:

Perform a final check to ensure the entire process aligns with the defined standards.

21. Document the results using the Bridge Concept Template.
22. Transpose the Bridge Concept in .md for storage.
23. Implement the Bridge Concept in a machine readable environment, following the technical specifications.

3.3 Bridge Concepts Candidate Term Selection

3.3.1 On the Importance of Candidate Term Selection

The meticulous **selection of candidate Bridge Concept terms** stands as a linchpin in ensuring the **success of the overarching methodology**. It becomes readily apparent that the choice of inadequate candidates carries the potential to significantly impede the workflow and the scalability of the tool. A list of possible issues is provided in what follows. It is imperative to underscore that the ensuing list is by no means exhaustive, and that further issues might emerge in full-scale field-testing.

To begin, it is paramount to acknowledge that an ill-considered choice during candidate term selection can lead to complications during the subsequent phases of Bridge Concept engineering. A candidate term fraught with issues may force an engineering team to resort to shifting the goalposts or, in extreme cases, to encounter protracted delays, wasting resources.

Moreover, the impact of a Bridge Concept constructed around an unsuitable candidate term can extend to the realm of its effectiveness in harmonizing ontologies and contributing to the OntoCommons EcoSystem. For instance, it may yield a limited number of connections or support links bereft of informativeness. The substantial **resource investment** inherent in the **Bridge Concept engineering process** (one of the drawbacks of the tool, counterbalanced by the openness of the recruitment pool) renders it inadvisable to entertain suboptimal choices. It should be noted that assessing the comparative efficacy and quality of Bridge Concepts may prove to be a complex undertaking, given the numerous variables at play. Ideally, it is imperative to anticipate worst-case scenarios (i.e., quasi-inert Bridge Concepts) during the engineering phase. Careful precautions should be exercised to mitigate the emergence of such scenarios.

A concomitant issue pertains to redundancy. In instances where two candidates are selected such that the resultant Bridge Concepts are likely to be closely connected or overlap completely (given an extension-centric perspective), it is apt to speak of **redundancy**. This circumstance can also arise when it is possible to establish a robust semantic relationship between a pair of Bridge Concepts, and one of them contributes little additional value over the other (i.e., the supported links are comparatively less informative or situated at a less important juncture in the network).

Since, in practice, Bridge Concepts will be engineered by different groups, having different goals, it is pivotal to maximize FAIR-ness, and reuse above all. Specifically, to mitigate the likelihood of non-trivial resource wastage, including the separate engineering of practically equivalent Bridge Concepts by distinct teams, there is a compelling need to foster active collaboration among the engineering teams, provide updated lists of the accepted Bridge Concepts and candidate terms considered for engineering, and organize a **centralized system for the evaluation of Bridge Concepts**.

Lastly, the selection of an inadequate candidate term may lead to the development of a Bridge Concept that, while pragmatically useful for aligning ontologies and contributing to the OCES, falls short of satisfying all the desiderata related to their roles in the framework. On a large scale, this could negatively affect the usability of the OCES, and compromise the broader standardization endeavor. It is essential to note, however, that this scenario may well be a far-fetched theoretical hypothesis, as it appears counterintuitive that a Bridge Concept, effectively connecting ontologies and devoid of issues in its engineering phase, would be rendered faulty by the candidate term around which it was constructed. Furthermore, in a scenario where the majority of Bridge Concepts meet the established desiderata, the presence of a few exceptions should not pose a threat to the EcoSystem's founding principles, nor should it stand out unduly.

In light of the aforementioned considerations, it becomes evident that the phase of candidate selection should not be treated lightly. Yet, given the number of possible scenarios, it might be questioned whether a singular, universal methodology should be followed (and whether that is feasible). In the context of WP2 and WP3 different approaches were considered, to test different possibilities, and the following guidelines are partially based on that. However, large-scale testing is needed. In general these guidelines offer practical suggestions and a minimally viable approach; however, this aspect of Bridge Concept engineering should be investigated more in the future.

It is worth reminding that there are obvious advantages inherent in standardization, and the adoption of a shared methodology ensures methodological transparency and augments the homogeneity of results, thereby making a significant contribution to the realization of interoperability and the enhancement of documentation accessibility. The principle of **standardization** holds a central position within the framework of OntoCommons, making it both prudent and fitting to have a common methodology rooted in FAIR principles, albeit one that welcomes **pluralistic interpretations**.

3.3.1.1 Variables in Candidate Terms Selection

In the context of candidate term selection, it is imperative to consider several **key variables**. Different scenarios necessitate distinct methodologies, each tailored to address specific challenges and amenable to the adoption of different tools and approaches. Candidate terms should be chosen after a **preemptive analysis and partitioning of the landscape**, to improve the results. The details of this endeavor are beyond the scope of this deliverable, but the reader can refer to the article cited above (de Baas, forthcoming) and D2.6.

Arguably, the paramount factor in this selection process is the sheer quantity of extant ontologies, as previously elucidated. The application of statistical analysis is rendered unreliable or, in certain cases, entirely inapplicable when dealing with a limited sample of ontologies. Conversely, a proliferation of ontologies introduces substantial scalability concerns, which exert substantial influence on the depth of exploration into individual concepts and ontologies.

Additionally, one must take into account the mean average number of concepts or entities present within each ontology. In scenarios involving expansive datasets, concerns pertaining to manageability by means of purely manual approaches become increasingly pronounced, while smaller datasets are better suited for thorough analysis.

As a general guideline, it should be anticipated that mathematical tools, when judiciously applied, yield comparatively superior results when addressing ontologies characterized by intricate architectures. Furthermore, the variance within these architectures represents another pivotal variable. It should be noted that certain methodologies may not be effectively applied to ontologies with specific architectural characteristics.

Moving forward, the identification of potential interconnections among the considered ontologies during the recognition phase assumes significant importance. As earlier emphasized, when these connections elude traditional analytical methods, the complexity of the endeavor escalates, necessitating a degree of creative problem-solving. It is important to acknowledge that standard scenarios are seldom found at the extremes, thus underscoring the pivotal role of this variable in shaping the relative prioritization of methodological steps, contingent on the strategic deployment of appropriate countermeasures.

Lastly, if the landscape analysis brought about the necessity to create new ontologies due to coverage gaps, the primary focus must not be exclusively, or even predominantly, on pre-existing ontologies and the concepts they employ. While these can indeed serve as guiding beacons for non-specialized candidate terms, the criticality of this approach is underscored when no relevant existing ontologies align with the requisites and desiderata of the Consortium.

The core variables are schematically reported in what follows for the sake of simplicity:

- The number of existing ontologies involved.
- The average number of entities per ontology.
- Whether tentative connections among the considered ontologies are established upon initial recognition.
- Vocabulary/standards coverage within the relevant domain.

As a caveat, the following sections will focus on the identification of candidate terms for the engineering of classes-Bridge Concepts, for the sake of simplicity, and given the priority assigned to them by the Consortium during the project.

3.3.1.2 Standard Approach Workflow: n. of Ontologies ≤ 3

The Consortium developed **two strategies** for the analysis of candidate terms based on the factors mentioned above. These strategies served as guidelines and were adapted to specific scenarios by focus groups in the context of WP3, with relevant adjustments.

The first methodology, detailed below, is particularly relevant for **scenarios involving a limited number of existing ontologies** (potentially due to gaps in the chosen domain's ontology-coverage). It hinges on **manual procedures** and entails a substantial reliance on **supplementary tools such as glossaries and golden standards** to fill in gaps when necessary. Notably, mathematical tools are unsuitable for the scenarios addressed here, necessitating active involvement from ontologists and domain experts. This approach allows for an in-depth analysis of existing ontologies, with the objectives of understanding how the domain is typically represented through ontologies and establishing tentative semantic connections.

The main steps are reported below for practical purposes; details will be offered in what follows.

1. Analysis of the aptness of Existing Bridge Concepts to avoid redundancies (since this section focuses on candidate terms selection, relevant considerations will be omitted/taken for granted in what follows).
2. Manual analysis of the involved ontologies, leading to the establishment of highly tentative and rudimentary semantic alignments.
3. A comprehensive comparative evaluation of the fundamental concepts that underlie the tentative semantic alignments. This process considers the suitability of terms to support the engineering of effective bridge concepts, culminating in the compilation of an initial list of potential candidates.
4. A reexamination of the branches within the target existing ontologies, particularly those not covered by the initial potential candidates. This reevaluation adopts a creative approach aimed at identifying concepts that may serve as superclasses.
5. The analysis of golden standards and glossaries, accompanied by active collaboration with domain experts, to identify additional candidates.
6. The final selection of candidates from those identified in steps 1 through 4.

The primary task is the selection of candidate terms for the subsequent development of Bridge-Concepts. This entails considering the concepts underpinning the labels and their usage in the source ontologies. The first step thus involves finding commonalities among the concepts in the various ontologies. Strategies for establishing tentative semantic alignments supporting terms-clustering in this context largely align with standard ontological practices. Developers from the relevant ontologies should ideally participate to clarify their stances and decisions, solving issues resulting from possible lack of explicit documentation (a common phenomenon) and interpretative difficulties.

Particularly noteworthy is the role of partial mappings among TLOs, a tool available exclusively within the OntoCommons Project. This resource can significantly aid in the identification of potential candidate terms by offering constraints and reference points for clustering. It should be emphasized that these tentative semantic alignments are intentionally kept rudimentary. The aim is not to create exhaustive alignments but to identify recurring (families of) **concepts** and gain a broad understanding of ontology architectures, with a focus on possible dependencies and connections. Again, it is worth pointing out that recurring **terms** should not be disregarded despite their opaqueness and can offer important hints.

Assuming that Step 1 identifies potential links among existing ontologies, the next challenge is to assess whether the prospective Bridge-Concepts, to be engineered starting from the identified concepts (typically those deemed equivalent to or superclasses of concepts in other ontologies, but that depends on the prospected use), can fulfill their intended role within the framework. This evaluation is necessarily holistic and **heavily relies on the comprehensive knowledge, expertise, and intuition of ontologists and domain experts**.

Steps 1 and 2 are best viewed as a collaborative negotiation process between ontologists and domain experts. Although their focuses may differ, they ultimately contribute to the same end goal. Ontologists often emphasize contextual aspects related to meaningful ontology connections, whereas domain experts prioritize the selection of candidate terms in alignment with relevant literature and golden standards. It is important to emphasize that the actual use of given classes in the ontologies should be kept as a point of reference, as there might be a noteworthy gap between theory and practical implementation.

With the list of candidate terms compiled in Step 2, it's reasonable to assess whether the prospected Bridge Concepts would offer sufficient concept-coverage to ground alignments among the selected set of target ontologies (for a start, and for generic relevant knowledge representation artifacts in general). While the actual Bridge Concepts have yet to be developed, the list of candidate terms should provide an initial gauge of task progress. In some instances, it may become evident that a concept from an ontology, along with its subclasses, would not be equivalent to or subclasses of any of the resulting Bridge-Concepts, as none of the candidate terms appear related to that specific semantic area.

It is then the responsibility of domain experts to determine whether integration of the relevant ontology branch is unnecessary, desirable, or mandatory. Meanwhile, ontologists should evaluate coverage from an abstract perspective. If necessary, the engineering team should reanalyze the existing ontologies, adopting a creative and open approach. Even if the ontologies do not provide immediate solutions, it may be possible to identify superclasses or weak external connections among ontologies, which can be highly beneficial in the OntoCommons Expert System (OCES) context.

Step 4 becomes crucial when Steps 1 to 3 yield suboptimal results, potentially leading to an empty list of candidates. In cases where existing ontologies cannot fully determine the list of candidate terms, other sources must be employed. Given the overall approach, glossaries, golden standards, and widely accepted resources present suitable alternatives. The selection of primary sources and specific terms should be guided by domain experts, who leverage their knowledge of the domain and related materials and tools. In fact, they may also suggest scenario-specific candidate terms directly, without strict reliance on domain-specific resources.

All the terms selected in the previous steps must be then aggregated and compared to make the final selection of candidate terms, potentially complete of a ranking with clear priorities. At this stage, any redundancies should be addressed. It is advisable to double-check whether the final projected ontology/domain coverage is satisfactory. Additionally, documenting the choices and underlying rationale is recommended for the sake of transparency and to avoid deviations from the original plans and rationale during subsequent engineering phases.

3.3.1.3 Statistically Grounded Workflow (experimental): n. of Ontologies > 3

In this section, the **second methodology** for selecting candidate terms is presented. This methodology is tailored to scenarios involving a **substantial number of existing ontologies**. As previously mentioned, this approach is **semi-automatic and heavily relies on statistical tools and network/data analysis**. It is crucial to note that the approach suffers from **lack of effective field testing**, although a simplified version of this strategy was employed in the already cited de Baas et al, forthcoming. This is particularly relevant for what concerns the formulas offered infra, which should be understood as merely indicative (especially when it concerns the proposed coefficients/weights): they are reported for the sake of completeness and to offer a basis for further development. (Notably, it might be also possible to make use of AI-grounded methods to optimize said parameters depending on the exact characteristics of the scenario to be tackled.)

The main steps are reported below for practical purposes. This strategy is organized in two Phases, covering Data Analysis and Integration, respectively. Details will be offered in what follows.

1. Analysis of the aptness of Existing Bridge Concepts to avoid redundancies (since this section focuses on candidate terms selection, relevant considerations will be omitted/taken for granted in what follows).
   b. PHASE A: Data Analysis
2. Standardization of the existing ontologies' architecture: one, and only one, root class/node/concept per ontology.
3. Application of an automatic mapper not focusing merely on terminological similarities with a demanding connection-establishment threshold.
4. Evaluation and preliminary validation of the mappings produced automatically, conductive to the identification of potential candidates to be weighted and compared.
5. Analysis of the average characteristics of the potential candidates: number and distribution of the sub classes & node location in the ontologies.
6. Calculation of the final value of the candidate (classes of) terms; explicit selection of a representative term and creation of a list given the ranking system.
   b. PHASE B: Integration
7. Refinement (and potential integration) of the automatic selection by domain experts.

In the first step, the selected ontologies first undergo a standardized formatting to enable the application of the necessary mathematical tools. The majority of ontologies already possess a single root class or concept, and thus require no adjustments. However, for those that deviate from this norm, an artificial root class is introduced, and all other classes within the ontology become its subclasses. This methodology transforms the ontologies into directed graphs or trees, ensuring a comprehensive interconnection of all classes and nodes. This interconnection is established through direct sub-class or super-class relations, as we will assume throughout this process, unless explicitly stated otherwise. Node distance calculations are based on this standardized format.

It is crucial to emphasize that, for the analysis conducted in Phase I, the ontologies are considered as standalone entities, irrespective of their potential connections to higher-level ontologies or their modular nature. This isolation of ontologies for analysis purposes is a pragmatic approach to cope with scalability constraints.

Once again, relying solely on a simple statistical analysis based on term or label occurrence proves inadequate, given that terms often lack transparency regarding the underlying concepts. In fact, terminological standardization remains an ongoing challenge within the project. The complexity of the existing ontologies also prohibits conducting an in-depth analysis, further emphasizing the importance of scalability considerations.

To address these challenges, the Consortium proposes the **repurposing of automatic mapping** tools to identify tentative connections among the ontologies' entities. In selecting a mapping tool, it is crucial to prioritize ones which are not merely based on terminological approaches (that would be beyond the point). For the stated purposes, it is also preferable to err on the side of false negatives rather than false positives. Therefore, selecting or configuring a tool with a high score threshold for alignment, whether for class equivalence or asymmetric relations such as sub-class and super-class, is deemed more appropriate. By imposing a demanding alignment threshold, it is ensured that only a manageable number of connections emerge, unless significant commonalities exist among the existing ontologies. Subsequently, the automatic alignments can be manually evaluated and potentially validated by domain experts and ontologists.

During this step, potential inconsistencies should be diligently addressed. For instance, if the automatic tool suggests that term A is equivalent to term B, term B is equivalent to term C, yet term A is not equivalent to term C, an evident inconsistency requires resolution.

Equivalence classes identify a single proto-candidate, representing a class of terms from which one can be chosen. Sub-class relations are also crucial for our objectives. As such, any discrepancies or inconsistencies in sub-class relations should be manually rectified. As a general principle, sub-class relations should be consistently distributed across classes of equivalent terms, aligning with the "sub-class of one, sub-class of all" approach. This principle is in harmony with the choices made when selecting and configuring the automatic mapping tool.

A term qualifies as a potential candidate if, at the conclusion of the mapping and validation procedure, it satisfies either of the following criteria: (1) it belongs to a class of terms deemed equivalent or (2) it stands for a class C in ontology O, such that there is a class C′ in an ontology O′ different from O and C is (tentatively) a superclass of C′.

The baseline value of a candidate is determined through the following formula:

$$BASE = \frac{(n) + \frac{m}{3}}{t}$$

whereas n is the number of ontologies in which the class of terms appears given the alignments (one, i.e. the one of provenance, if no class-equivalences were individuated); m is the number of ontologies containing sub classes of the class of terms and in which members of the equivalence class do not appear; and t is the total number of ontologies under investigation.

At most, the BASE value will be 1, while the lower boundary is $\frac{1}{t}$. Giving a baseline value to all terms should not be problematic since the formula should be applied only in cases involving automatic mappings. Sub class relations are tentatively considered given minimal testing. This parameter should be adjusted if found to give incorrect results, and possibly made dependent on characteristics of the involved ontologies.

The option of utilizing multiple automatic mapping tools to streamline and automate the validation process is worth considering, although an exhaustive discussion of this topic is beyond the scope of this document.

In theory, one could conclude the selection process upon reaching the BASE value, subsequently establishing a ranking before advancing to the second phase. However, exercising prudence dictates a refined selection process that aims to pinpoint terms conducive to the development of valuable Bridge-Concepts. This refinement is of paramount importance, given the substantial potential for harnessing data analysis. Valuable insights from differentiation of Bridge-Concepts into two categories (depending on whether they are mostly focused on establishing vertical or horizontal alignments, see §3.1) may provide valuable guidance in this regard.

To underscore key determinants, the connection of a Bridge-Concept with a specific ontology is pivotal in ascertaining its role and potential. Nonetheless, the architecture of the target ontology significantly influences this connection. Equally crucial, as a general guideline, are network connections. These connections play a pivotal role in the effective dissemination of reasoning and the facilitation of data sharing. It is imperative to recognize that aspects concerning data sharing transcend mere abstract properties. While the frequency of term occurrence, a central aspect of Step 3, indeed identifies critical junctures for data sharing, there remains room for further refinement in this domain.

Taking these considerations into account, let us define the network coefficient NK of a certain candidate (class of terms/term) for a given ontology in the following way:

$$NK = \sum_{d=1}^{e} \left(\frac{f(d)}{c} \right) \times \frac{1}{d}$$

whereas (in a certain ontology, let us call it O_1) $f(d)$ is the number of the sub classes at a certain node-distance d (and thus a function of d); e is the maximal distance of a sub class from the candidate in O_1; and c is the total number of classes belonging to O_1 (all the parameters obviously depend on the architecture of the ontology considered).

NK depends on the number of sub classes, taking into account the distance: the underlying rational being that node-distance is inversely proportional to the specificity of reasoning and the how meaningful the resulting connections are to the end of data sharing.

However, the primary concern lies in calculating the average for all the ontologies in which the candidate possesses sub-classes, specifically in those ontologies where the candidate is included:

$$ANK = \frac{NK_{O_1} + NK_{O_2} + \ldots NK_{O_n}}{n}$$

Evidently ANK imposes significant penalties on meaningful ontology connections, which may, nevertheless, be of great importance for the objectives at hand. This issue can be addressed by introducing an additional parameter based on node position, the weighting of which should also serve to balance the fact that specialized terms are statistically less likely to appear in numerous ontologies:

$$POS = \frac{x}{y}$$

whereas x is the distance from the root for a given ontology, and y the maximal distance of a sub class of the candidate (or of the candidate, if it has no proper sub classes) from the root of the ontology.

Once again, the average $APOS$ has to be calculated, taking into account all the ontologies a candidate appears in.

The formula to determine a certain potential candidate term value is then the following:

$$Value = BASE^2 \times (ANK + APOS)$$

All pertinent variables are readily accessible and easily obtainable using standard ontology development and analysis software. As anticipated, it is essential to emphasize that this approach has undergone limited testing, and its practical efficacy remains to be determined. It is possible that certain parameters may require adjustments, or the approach itself may need fine-tuning to align with the specific characteristics of the ontologies under consideration, including the potential addition of further parameters.

This step concludes PHASE I. A PHASE II is however required, as the ranking of potential candidates based on the procedure outlined in Phase I should not be considered as definitive - but rather as a guiding reference. Upon generating the list, the final selection of candidate terms is entrusted to domain experts who will concentrate on requirements that an automated process, especially one as rudimentary as the one described above, may not have adequately addressed.

This approach ensures a thorough evaluation of results, including the identification and removal of subpar selections and the rectification of minor deficiencies within the selection algorithm. It is noteworthy that certain terms, though not directly pertinent to the domain under examination, are anticipated to appear across multiple ontologies due to their practical applications.

Upon completion of this stage, presuming that there are no remaining gaps to be addressed within the scenarios covered by this second approach, a definitive list of candidate terms can be compiled.

3.3.2 Characterizing Bridge Concepts

Bridge Concepts are **chiefly (informally) characterized through their elucidations**, but also via labels, the specification of the more relevant domain, as well as via the connections with domain knowledge resources and existing ontologies' concepts, which should explicitly discussed in the documentation.

Bridge-Concepts' elucidations describe how they connect to the world, and are the main **indication of their usage** as tools in ontologies. Notably, elucidations are **not exactly definitions**, as they are meant to respond to practical concerns. In general, they:

- must exhibit a balance between flexibility (focus on intuitiveness assuming background knowledge, rather than on a set of formal constraints which might appear obscure, or which might be deemed non-core according to approaches embraces by different stakeholders)
- and rigidity (pragmatically well-defined extensional boundaries, necessary for machine-implementation given the impossibility of bargaining in conversation),
- maintain strong links with primary Knowledge Domain Resources
- and ontology entities which they are meant to connect,
- and align with common sense.

In order to achieve the desired fine-graininess, Bridge Concepts' elucidations should **specifically address ambiguities** which are relevant given the prospected usage. Notably, making "value gaps" can be informative in this context, e.g., specifying that certain traits are not relevant as a concept is more coarse grained.

Sometimes it might be useful to offer specific examples and counterexamples, where it is deemed necessary. However, it is better to avoid doing this if general principles can be provided, to avoid prototyping effects, and when the concepts are too specific.

In addition to these criteria, it is possible to introduce further desiderata. To ensure usability for domain experts, Bridge-Concepts should **abstain from referencing ontological entities** that are characterized within a specific ontology, and specifically to aspects related to their formal characterizations. This represents the first dimension of neutrality: **ontology neutrality**.

In their role as bridges between various worldviews, especially among different TLOs and MLOs, Bridge-Concepts should **minimize unwarranted ontological commitments**. However, complete ontological neutrality remains an unattainable ideal, and pragmatism should guide any necessary commitments. More specifically, ontological neutrality translates to the principle of **avoiding reference to traits that are not inherent to the specific domain chosen for elucidation**.

Furthermore, **brevity** is a desired characteristic, although achieving a harmonious balance among these various considerations is a complex task. A complex elucidation might not result intuitive, and increases the risks of readers' missing core points. This requires an iterative process of refinement and adjustment.

In general, elucidations should be organized in the following way:

1. Brief Introduction making use of domain experts' background knowledge.
2. Informal description with implicit references to selected Golden Standards and Ontology entities (the description should closely follow them).
3. Note on use of the concept in the domain.
4. Resolution of ambiguities via traits and values commonly referred to.
5. Possible examples and counterexamples.

The above list should be understood as a recommendation, rather than a requirement. The various parts can easily be identified in the following example, save for (5), in line with the discussion above:

GENERAL CONCEPT INFO:

IRI:	*Suggested entity new IRI.*
OWL Type:	*Class*
Concept Elucidation:	An atom is a nucleus surrounded by an electron cloud. The nucleus consists of electrically positive protons and electrically neutral neutrons, and carries almost all of the atom's mass; the electron cloud is a quantum system made of one or more bounded electrons, and is pivotal in determining the atom's size and properties. It is the smallest system that has the characteristic properties of a chemical elements and, as such, it is often employed as a unit in the domain of chemistry. Atoms can either be standalone or bonded; they can have an unbalanced number of electrons with respect to their atomic number (the latter being determined by the number of protons in the nucleus) or have a net electric charge. Domain: Natural sciences - Physics / Chemistry.
Labels:	*Labels used to address the concept, ordered as:* *skos:prefLabel: Atom* *skos:altLabel: Atom (Broad)* *skos:hiddenLabel: Chemical Element; Neutral-or-Ion Atom; Standalone-or-Bonded Atom*

Figure 3 - Example: Bridge Concept Elucidation

The **preferred label** is especially important, since it is the first, and most salient, element impacting a user, thus its choice is especially important. As such, it should be defined last in the Bridge Concept engineering process. In some cases, it might be useful to sacrifice immediateness to avoid possible misunderstandings.

Finally, the **comments** discussing the connections, and divergences, relative to knowledge domains resources, and existing ontologies' concepts (once the Bridge Concept is aligned to them) should be extensive. Not only this can greatly improve Bridge Concepts' FAIR-ness, but it is worth keeping in mind that standards act as a point of reference, and that some stakeholders might find these comments more illuminating than the elucidations themselves.

One notable feature of Bridge Concepts is their potential to serve as a standardized Vocabulary, alongside their primary function in ontology usage. While primarily designed for ontology applications, they offer the prospect of evolving into a recognized Standard if they prove valuable.

3.3.3 Aligning Bridge Concepts: General Principles and Guidelines

Bridge Concepts establish **mediated semantic connections** among ontologies, and specifically among concepts belonging to different ontologies which, in standard scenarios, will be characterized in different ways, both formally and informally. The Bridge Concept alignment methodology focuses on **semiotic interoperability**.

Bridge Concepts are not aimed to connect classes/concepts which are intensionally equivalent or such that subsumption under one entails subsumption under the other in all possible worlds (more complex axiomatic relationships can also be considered). As per their name, they should be understood as bridges allowing practical connections despite the relata's idiosyncrasies, rather than as the elicitation of relations de facto, or as harmonizing tools apt to create a rigid unified system via ontology integration. In fact, a unificatory approach would stand in contrast with the project's conservative strategy, which refrains from endorsing integration activities and instead embraces pluralism.

Bridge Concepts' focus is on **extensionality**, and thus on **individuals**. Adding onto that, and in line with the background theoretical assumptions, the tool just needs to be **as precise as needed in order not to cause practical issues**: the presence of borderline cases and exceptions needn't undermine the **connections' informativeness** and **practical usefulness in certain contexts**. In turn, if there are cases causing frictions and interoperability issues, they cannot be recognized as borderline anymore, and the established connection is not effective (again, in a given context).

As a general point extending beyond Bridge Concepts themselves, from a merely practical point of view, ontology alignment aims to establish connections that facilitate the **seamless transition between distinct representational systems**. To illustrate this point: let S be a generic scenario being modelled, O′ and O′′ two ontologies and M′ a set of mappings from O′ to O′′, and R′ and R′′ the ways in which S would be modelled in O′ and O′′ respectively; a perfect mapping is such that for any generic scenario S, R′ + M′ fully recovers R′′. Obviously, there are **inherent limitations due to**

the ontologies' different domains of applications/perspectives/commitments, so this can be considered as an ideal determining the scale of alignments' informativeness.

Bridge Concepts attempt to achieve this aim **locally**, and, specifically, at **junctures which maximize the transmission of informative data**. **Semantic interoperability** is then achieved through the establishment of mediated semantic connections. These connections may ultimately manifest as complex axioms, transcending the scope of simple taxonomical inclusion relations. Bridge Concepts also go ways towards **addressing terminological heterogeneity**, by providing easily findable labels (and support for queries) and through the mediation of the semantic relations established. This **mediation serves to clarify the labels**, facilitating the differentiation between synonyms and "false friends."

The focus on prospective usage introduces inherent challenges, as diverse stakeholders are likely to employ the framework for distinct purposes. Bridge Concepts have to be **tailored to specific precision levels**, and different Bridge Concepts (more or less fine-grained) might have to be introduced to accommodate different needs. In turn, Bridge Concepts contribute to the establishment of a layered and adaptable environment.

With these preliminary theoretical considerations out of the way, it is possible to define a list of recommendation specific for Bridge Concepts alignment:

- Bridge Concepts ought to be aligned with at least 1 class from each TLO part of the TRO.
- If the target ontology is part of the TRO, the alignments should be consistent with the alignments among TLOs.
- If the target ontology is not part of the TRO, the alignments should be consistent with those that would be made from the target ontology to the TRO.
- Bridge Concepts should ideally have at least a subclass and a superclass (or an equivalent class) in the target ontology as a result of the alignment.
- Alignment should be as informative as possible.
- Alignment should be consistent across the framework; if they aren't, the matter should be investigated.
- Priority should be given to the formal characterization of target ontology entities in the alignment.
- Given lack of documentation or under determinate scenarios, contextual information should be taken into account (how certain classes are used) and safe alignments should be privileged.

3.3.4 Concept Clusters

In some specific cases it might be beneficial to **cover specific areas of the conceptual landscape more thoroughly** to better avoid ambiguities and effectively ground a **multi-layered approach**. Instead of standalone concepts, it is possible to introduce **Concept Clusters**: sets of semantically connected Bridge Concepts. Concepts Clusters divide the logical space identified by an **apical Bridge Concept** via exhaustive and exclusive subclasses based on a **list of** easy-to-grasp and possibly intuitive **traits**, making use of a **frame-theoretic approach** tacitly assumed by many ontologies.

This option expands the methodology to non-degenerate **Ontology Content Patterns**, and towards ad-hoc mini-ontological modules. The pyramidal organization of these clusters reflects a series of **yes-no questions** defined by the list of traits, which guides users in assigning individuals/data to the most appropriate category, in order to avoid mistakes.

Concept Clusters also make it possible to establish more informative connections even without direct data pipelines, with the individuation of superclasses common to concepts in different ontologies which are somehow related, but such that no strong semantic relationship can be established among them.

As a caveat, experiments with Concept Clusters and lists of traits were limited in the context of the Project, and it is extremely important not to exceed in their use to **avoid creating out and out ontologies**, losing the benefits of Bridge Concepts' plasticity and modularity.

3.3.5 Maintaining the MRO

The initial MLOs and DLOs selected for inclusion in the OCES were chosen by the Consortium based on well-reasoned parameters, consultations with stakeholders, and an analysis of the landscape (see D2.2, D3.2, D6.2 among others). Many of the points discussed regarding the maintenance of the TRO are equally relevant, mutatis mutandis, for the Mid-Level Reference Ontology (MRO).

As repeatedly emphasized, the OCES framework is **open** and requires **minimal entry investments**, especially for MLOs and DLOs. Existing ontologies simply need to be connected to the relevant Bridge Concepts to become integrated into the entirety of the OCES. Additionally, where feasible, it is **recommended to establish robust subsumptive alignments with at least one TLO**, following the LOT methodology endorsed and recommended by OntoCommons. This approach enhances the number of alignment checks, ensuring enhanced practical interoperability.

Creating new ontologies aligned with the entire EcoSystem is likewise fairly simple, as stakeholders can ground their ontology on an offer of different TLOs, and reuse modules of existing ontologies. Including classes statedly equivalent to Bridge Concepts, they can greatly simplify, or even sidestep completely, the alignment stage, establishing maximal informative links/data pipelines in the process.

Updates in MLO and DLO ontologies have a less disruptive impact on the framework compared to changes in the TRO. The framework's modularity, as discussed in §5.2.2, facilitates the handling of even significant changes. Furthermore, multiple versions of an ontology can be included in the ecosystem without adverse effects if they are actively used in the field. The targeted spot-links facilitated by Bridge Concepts significantly reduce the costs associated with re-alignment and alignment maintenance, although the inherent complexity of the procedure should not be underestimated.

Regarding the maintenance of Bridge Concepts themselves, it is crucial to **incorporate new knowledge domain resources into the documentation**. Care should be taken to **avoid shifting the intended informal characterization** as much as possible. Similarly, elucidation should only be adjusted to significantly enhance accessibility for domain experts.

Given their inherent flexibility within pragmatic limits and a strong belief that Bridge Concepts should not become rapidly obsolete, field testing is required to confirm this. In general, it is advisable to introduce new Bridge Concepts and potentially replace obsolete ones rather than attempting to adjust existing ones in response to shifts in the landscape. This approach helps prevent suboptimal alignments resulting from 'meaning-bargaining,' as such issues are particularly subtle and can significantly compromise the practical effectiveness of the framework.

3.4 Bibliographical References for §§ 3.x

- (Barsalou, 1992) Barsalou, L. W.: Frames, concepts, and conceptual fields. In A. Lehrer & E. F. Kittay (Eds.), Frames, fields, and contrasts (pp. 21–74). Hillsdale: Lawrence Erlbaum Associates.
- (Cappelen, 2018) Cappelen, H.: Fixing Language: An Essay on Conceptual Engineering, Oxford: Oxford University Press.
- (Carnap, 1956) Carnap, R.: Meaning and Necessity, Chicago: The University of Chicago Press; 2nd Edition.
- (Clark, 1996) Clark, H. H.: Using Language. Cambridge University Press
- (de Baas et al., forthcoming), de Baas, A.: Review and Alignment of Domain-Level Ontologies for Materials Science, IEEE Access.
- (Euzenat, 2001) Euzenat, J.: Towards a principled approach to semantic interoperability, OIS@IJCAI.
- (Gross & Yellen, 2005) Gross, J.L. & Yellen, J.: Graph Theory and Its Applications, CRC Press, 2nd Edition.
- (Leitgeb, 2023) Leitgeb, H.: Ramsification and Semantic Indeterminacy, The Review of Symbolic Logic, Vol. 16, issue 3: 900-951.
- (Melvin, 2022) Melvin, F.: "Intensional Logic", The Stanford Encyclopedia of Philosophy (Winter 2022 Edition), Edward N. Zalta & Uri Nodelman (eds.), URL = <https://plato.stanford.edu/archives/win2022/entries/logic-intensional/>.
- (Quine, 1960) Quine, W.v.O.: Word and Object, Cambridge, Mass.: M.I.T. Press.
- (Savić & Ivanović & Jain, 2019) Savić, M. Ivanović, M., Jain, L.C.: Complex Networks in Software, Knowledge, and Social Systems (Intelligent Systems Reference Library Book 148) 1st Edition.

4. OCES: Formal Framework

4.1 Introduction to the OCES Formal Framework

H2020 OntoCommons project broadly aims at developing Ontology Commons Ecosystem (OCES) harmonising data documentation through ontologies and taxonomies, making the data FAIR and enabling intra- and cross-domain interoperability. As OCES is the primary and most tangible outcome of OntoCommons project, it is important to detail the specification of OCES by identifying its components and exploitation strategies. Moreover, this document also includes several intermediate results and clear future directives as part of the specification of OCES for this document is being written in the month 18 of the OntoCommons project.

As part of objective 2 of OntoCommons project, OCES will consist of:

- An OntoCommons Top Refence Ontology (TRO), in the form of a selected set of existing and widely used Top Level Ontologies (TLO) made of a mutual set of alignments between the selected TLOs (i.e. axioms providing correspondences between entities of TLOs), that will formally constitute the apical point of the hierarchy. An ontology alignment approach (the so-called harmonisation) will maximize the use of existing domain ontologies developed under different TLOs.
- Middle Level Ontologies (MLO), to allow smooth connections between TLOs, lower-level ontologies and commonly needed entities such as time, information, unit, space etc.
- Domain Level Ontologies (DLO), as needed by demonstrators, both harmonised existing domain ontologies and newly developed domain ontologies, following the develop/test/validate/agree procedure.
- EcoSystem Requirements and Specifications, to ensure homogeneity between ontologies becoming part of the OES, such as formalization in specific ontology language and documentation.
- Tools, a selected set of tools for the practical implementation of data documentation and its exploitation, that are ready to be used with ontologies respecting OCES requirements.

In summary, OCES can be described as a combination of fully harmonised ontology artifacts and associated tools and methodology for building the existing and future ontologies. The arsenal of OCES therefore provides a complete solution for data documentation in the NMBP domains. Here we enlist the needs of data documentation which OCES provides a solution for in **Error! Reference source not found.**.

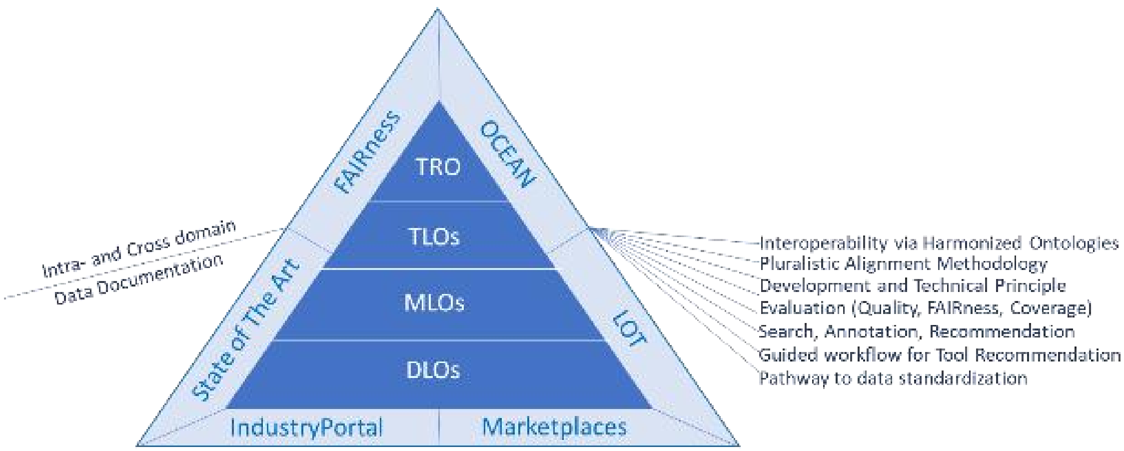

Figure 4 - OCES for providing solutions for needs in intra- and cross-domain data documentation

- **Interoperability of data via harmonised set of ontologies:** The hierarchical network of ontologies, which forms the core of the OCES, is stratified according to the different levels of generality from top to domain levels. The top-level ontologies (e.g., BFO, DOLCE, and EMMO) are aligned with rigorous semantic mappings (based on full first order logic proofs) that are stored in a top-reference ontology in loosely connected module providing users freedom to choose one or more TLOs for lower-level ontology development on demand. Various existing MLOs are vertically aligned to every TLOs via a set of bridge terms and, consequently, are also aligned among themselves. These upper-level ontologies provide a foundation for aligning the domain level ontologies as the alignment among them are guaranteed provided they reuse the concepts from the TLOs and MLOs which are aligned to the OCES. Specifically, the set of bridge terms in the MRO and the rigorous mapping from TRO are exploited to derive the cross-domain mapping among these DLOs. For sake of separation of concern, the common set of concepts to specific topic areas under materials and manufacturing will be abstracted into several domain reference ontologies for supplying the application specific ontologies with highly specific scope with a set of ready-to-go patterns for quick development and reuse.

- **Pluralistic alignment methodology:** The harmonised core of layered ontologies is not an ad-hoc one-time effort for harmonising the existing ontologies but envisioned to continuously grow as new ontologies from every level of generality are added to it. To ensure the continuation of this effort, OCES also charter a robust methodology by adopting a pluralist approach which promotes reuse of already available resources with more inclusive approach than monism. However, aligning ontologies built with different ontological commitments and allegiance to different top-level ontologies is extremely difficult. From the coordinated effort of work package (WP) 2 and 3, OntoCommons the technical details of such pluralistic alignment methodology (PAM) are designed. The PAM provides individual strategies specific to top level, mid-level, and domain-level ontologies concerning various aspects, e.g., rigorousness of alignment, types of alignment, modularisation, and encoding strategies. The importance of PAM is crucial for the sustainability of the exploitation of ontologies in documenting data for NMBP domains as it ensures the interoperability among these semantically annotated data is maintained by continuously incorporating the required semantic models into the harmonised ontology network of OCES.

- **Technical principles for ontology development**: Development of ontology for a specific subject or task needs to be consider many other aspects aside the alignments to other ontologies to be successful. These aspects include the requirement analysis, engineering, deployment, and maintenance. Currently, myriad of practices and protocols, both standardised to customised, are adopted by ontology developers as suited by their need. However, as recommended by several stakeholders and domain experts in past surveys, workshops, and focus group meetings, without such a clear and comprehensive guideline for technical development, management, and administration of ontology engineering activities, establishment, and sustenance of the harmonized ontology network of OCES will be pose a great challenge for not only the ongoing effort of OntoCommons members but also for the future adoptions. With adaptations, specific to the harmonization goal of OCES, an ontology engineering methodology with a foundation in Linked Open Terms (LOT) is prepared. This methodology includes detail steps for extracting and analyzing ontology requirements from use cases and subject matter experts using proven techniques, e.g., competency questions; ontology building workflow as a practical guide for PAM as well as various concerns, e.g., encoding languages (e.g., CASL, OWL Lite/DL/Full etc.), reasoning tools (e.g., Pellet, Hermit, Fact++), serialization formats (e.g., OWL/XML, RDF/XML, JSON-LD, Turtle), mapping vocabulary, IRI and naming conventions, metadata, documentation strategies, as well as review and release management.

In the following sections, we briefly introduce the harmonised set of ontologies and the methodology for practical ontology development using this harmonised set of ontologies. In Section 5, the technical principle to concern many developmental aspects of ontology engineering is presented.

4.1.1 On the subsumption of Knowledge Representation Frameworks

Relational databases, Knowledge Graphs, and Lightweight and Foundational Ontologies all provide structured and methodical approaches for the modelling, storage, and management of data. While their functions complement one another, and are interconnected, there are notable distinctions among these technologies. It is reasonable to categorize them progressively based on the functionalities they offer, with a particular focus on loosely defined semantic aspects.

The Department of Administration (DoA) places a strong emphasis on ontologies as the preferred frameworks for knowledge representation within OntoCommons. Nevertheless, it is important to recognize that many stakeholders opt for simple databases as their primary data management solution. This choice can be influenced by various factors, including financial considerations and technology accessibility. In some instances, employing more intricate technologies may prove impractical, particularly in straightforward scenarios or when handling substantial volumes of data due to practical limitations.

Conversely, the incorporation of semantic enrichment plays a crucial role in mitigating error risks, augmenting data interpretability, and fostering data reusability, thus contributing to the enhancement of data capital. It is also instrumental in realizing interoperability beyond localized solutions. The quality of the mapping significantly impacts the enhancement of querying functionalities.

More specifically, the concern of interoperability may be focused on different level of knowledge engineering. As shown in Figure 5, users may target interoperability at any level, requiring all the interoperability levels underneath. For example, data interoperability at the semantic level requires syntactic level interoperability, that is data is modelled using the same syntax. Relational databases, and Knowledge Graphs, created without rich semantic only provides syntactic interoperability.

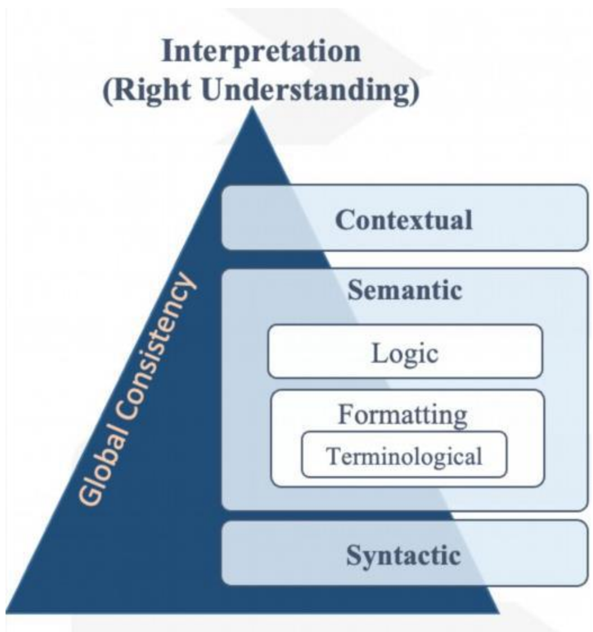

Figure 5 - Levels of Interoperability

Although, every ontology provides some level of semantic interoperability to the data modelled by it, the level of logical rigor as well good formatting and terminological work of an ontology greatly influences the quality of the interoperability in terms of consistency and interpretation, achieved by using that ontology. Therefore, ontology is not a silver bullet for solving data interoperability.

To accommodate these diverse perspectives, OntoCommons introduces a vertically layered approach. It offers stakeholders an interoperable network of ontologies (known as OCES) and tools to facilitate the mapping of their databases and knowledge graphs into these frameworks. In detail:

- Schema alignment is streamlined through one of the important mechanism of ontology alignment: Bridge Concepts (further elaborated below). This tool is designed to be user-friendly, making it accessible to domain experts and data scientists who may not possess specialized ontology knowledge. With the plethora of ontologies within the ecosystem and the introduction of bridge concepts, finding a suitable mapping becomes notably more straightforward for stakeholders compared to conventional methods.
- Comprehensive recommendations are provided for all data format transformation processes, including a clear set of requirements and best practices, ensuring data reusability for all stakeholders contributing to the EcoSystem, especially concerning annotations. The mappings themselves are built to be traceable, integrated, and adhere to FAIR principles (explained further below). This results in a semi-automatic workflow with straightforward steps.
- Semantic enrichment is guaranteed through the mappings themselves, implemented in the most expressive language. This approach takes into consideration OntoCommons' pluralistic philosophy and the richness of the EcoSystem, enabling informative characterizations and heightened customization.

As a result, OntoCommons effectively addresses key concerns related to costs and technology accessibility. It offers a service with manageable entry and maintenance costs while delivering various benefits. In practice, stakeholders can continue to use databases and/or knowledge graphs while leveraging the advantages mentioned above through alignment with OCES.

The decision to establish a framework that focuses on the most committed layer is also reinforced by the relatively straightforward process of reverting from ontologies to knowledge graphs and databases once the alignment has been established. Unlike translations into less expressive formal languages, which often lack a universally applicable algorithmic procedure, subtraction is generally more straightforward than addition.

4.2 OCES Formal Framework: Use and Maintenance

In this section, A detailed guide on using the OCES framework for practical ontology development is presented.

4.2.1 Description

OCES framework supports wide range of activities of ontology engineering. These activities are guided by corresponding methodological guide and a set of tools specially built for the purpose. The coverage of different methodologies and supports of tools for these activities are presented in **Error! Reference source not found.**.

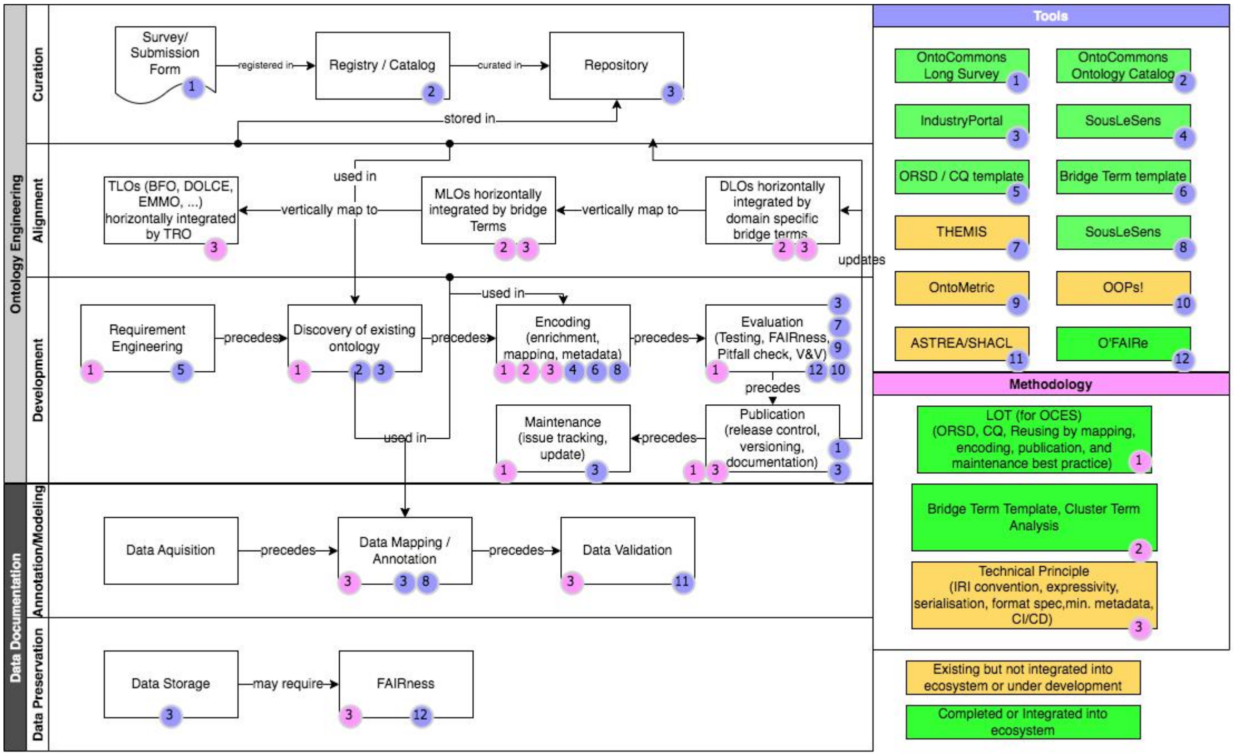

Figure 6 - Broad overview of different ontology engineering activities with tools and methodology coverage

4.2.2 Modularization and Inconsistencies

The core of OCES is composed of a hierarchy of networked ontologies of different levels of generality (from top-level to application level). As requested by the call text, a single apical ontology, called the *OntoCommons* **Top Reference Ontology (TRO)** will be provided to enable a common foundation for data interoperability between TLOs and lower level ontologies. The TRO will consist of i) a Meta Ontology (MO) and ii) a set of selected TLOs (i.e. BFO, DOLCE, EMMO). The MO will be the ontology used for aligning the selected TLOs, providing mapping between ontological. While TLOs provide a formal characterization of general concepts according to specific ontological perspectives, **Middle Level Ontologies (MLOs)** are primarily intended to extend those concepts towards a specific discipline (e.g. manufacturing, materials science, chemistry) with the aim to provide a core shared vocabulary for lower level modules. OCES organises the ontology stack in a way that not only facilitate user to flexibly adopt it for their need but also provide room for expansion and update in the future. Typically, the OCES stack of ontologies are ontology artefact encoded in OWL language. Despite the availability of the mappings among TLOs in FOL, the OCES stack only admits OWL version of the mappings. A representative portion of the stack is shown in **Error! Reference source not found.**. The modularisation is achieved with empty files (ontology artefacts containing only import statements but no term), which also helps in importing more than one ontology to be imported per module.

In this section the import structure of the OCES stack is described for resolving references in some DLO as such will require at least some ontology artefact to be imported from each level of generality. Although the ontology artefacts are physically stored in a folder structure closely following the import paths, please see Section 4.2.4 for the access mechanism of the OCES.

The tro file (http://www.ontolocommons.eu/tro) imports one tlo and optionally one meta module. Module tlo contain one or more TLO as import (it is performed by first importing them to an intermediate file called module called tlo (http://www.ontolocommons.eu/tlo). Note that it is not necessary to import all TLOs in the tro file. If more than one TLO is imported in the TRO file, then the mapping between those TLOs may also need to be imported in the TRO. The mapping files between TLOs are curated by meta file (http://www.ontolocommons.eu/tro/meta) which may import one or many mapping files as required. For example, if TRO file contains BFO and DOLCE then meta file should import bfo-dolce mapping file. The TRO file seldom needs to be changed for this purpose as tlo and meta modules can be used to handle the selection of TLOs. The TRO module helps the downstream modules to import one or many TLOs and their mappings without requiring to change the import statements if the TLO selection changes.

The bridge concepts act as anchors for the lower-level ontologies (core references and domain level) to top level ontologies. For this purpose, the mid-reference ontology (MRO) level of the OntoCommons EcoSystem (OCES) is founded on these bridge concepts. The 'bridge' module (http://www.ontolocommons.eu/mro/bridge) may contain one or more bridge concept files as required by the project. Every bridge concept file will contain mappings to some terms from each top-level ontology. The bridge module imports the TRO module containing only a selection of ontologies. Therefore, only those mappings to selected ontologies that are imported by TRO are active for each bridge concepts. For example, if the TRO contains only BFO, DOLCE, and their internal mappings, and the bridge module contains the bridge concept 'X', its mapping to a certain term emmo:Y will not have any effects on the reasoners as the TRO does not contain any the mappings that links emmo:Y to a term in BFO or DOLCE.

The bridge module, containing one or more bridge concepts, will be imported by a domain reference or domain ontology. The structure in Figure 7 shows an example in which they are imported by some domain reference ontologies, e.g., IOF-Core and CHEBI. The ontologies importing the bridge file will have their own organisation to embed mapping to the bridge concepts. These ontologies with their mapping to bridge concepts forms the mro (http://www.ontolocommons.eu/mro), then in turn can be imported by domain or application level ontologies for further reuse. The organisation provides a general guideline to structure any other project which may of course replace the host of the IRIs from http://www.ontolocommons.eu to their own.

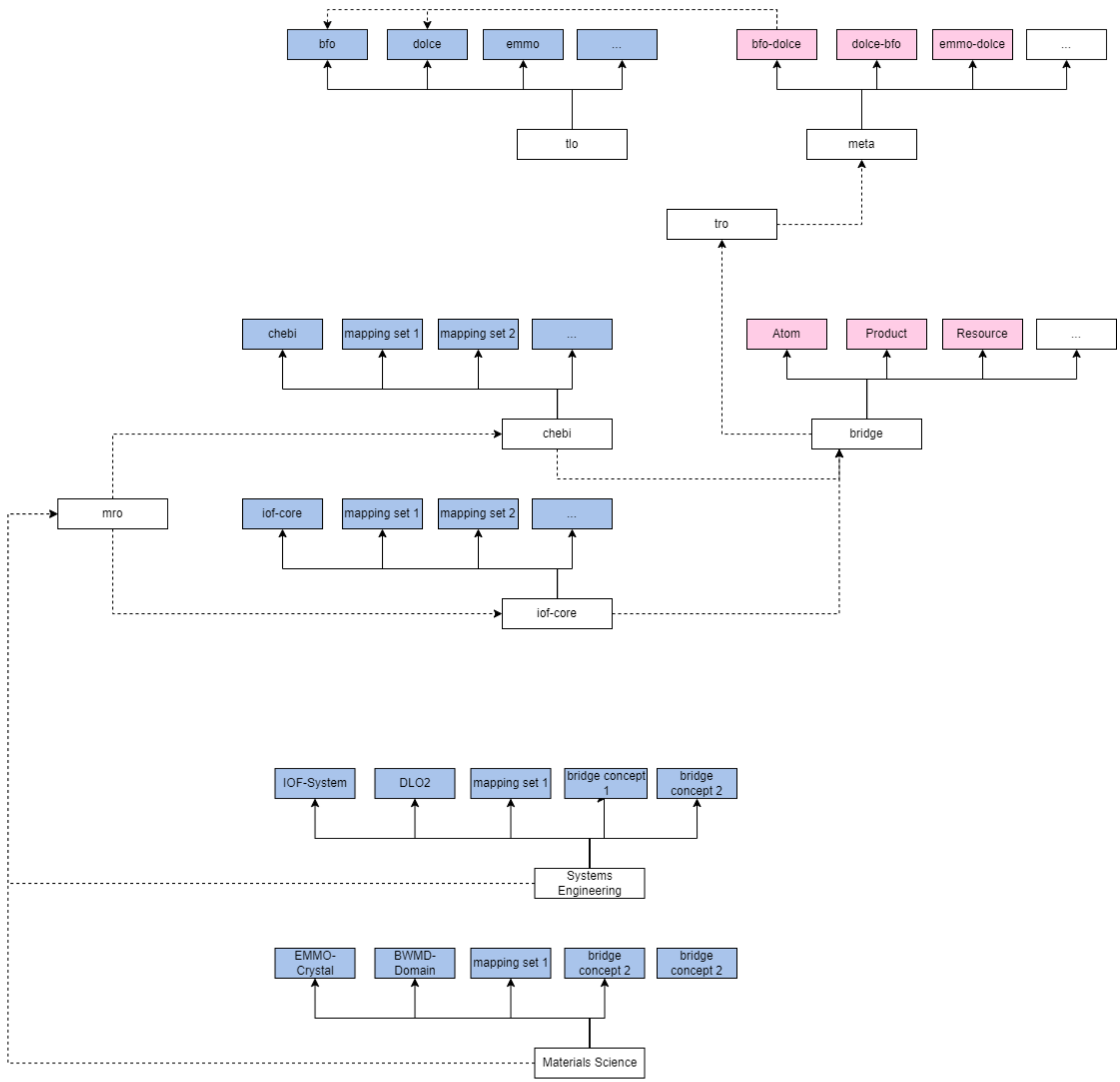

Figure 7 - Organisation and import structure of OntoCommons Ecosystem ontology stack (solid line denote the sharing of IRI and the dashed lines denotes import

As it has been discussed in §2.4, for the TRO, we followed a "safe" strategy to introduce OWL mappings. This strategy basically eliminates possible inconsistencies in the OWL version of the TRO (i.e., the union of the OWL versions of the TLOs and the OWL mappings).

However, (i) one can imagine to introduce less safe mappings that better approximate the FOL mappings but may generate inconsistencies in the TRO and (ii) it is not always possible to guarantee the same level of accuracy and safeness in the case of MLOs and DLOs included in the

OCES. For these reasons, it is important to have in mind some possible methods for "repairing" the OCES once it is affected by inconsistencies while preserving as much of the original knowledge as possible.

Classical methods to automatic repair inconsistencies in the context of Description Logics are based on removing axioms (Kalyanpur et al. 2005; Kalyanpur et al. 2006; Baader, Penaloza, and Suntisrivaraporn 2007). In doing that, these approaches often remove also intended implicit consequences, i.e., the removal of axioms solves the inconsistencies but at the same time, in several cases, removes also some wanted theorems that do not origin any inconsistency.

More recent approaches offer more fine-grained methods based on weakening axioms. Rather than removing axioms one makes them more general allowing for a less destructive approach able to preserve more of the original knowledge (Troquard et al. 2018; Porelle et al. 2018). Both these methods can in principle be used in order to manage inconsistencies in the OWL framework.

Relevant Bibliographical References for this section are reported in what follows:

- Baader, F.; Penaloza, R.; and Suntisrivaraporn, B. 2007. Pinpointing in the description logic EL+. In Proc. of KI 2007, volume 4667 of LNCS, 52–67. Springer.
- Kalyanpur, A.; Parsia, B.; Sirin, E.; and Hendler, J. 2005. Debugging unsatisfiable classes in OWL ontologies. Web Semantics: Science, Services and Agents on the World Wide Web 3(4):268–293.
  Kalyanpur, A.; Parsia, B.; Sirin, E.; and Grau, B. C. 2006. Repairing unsatisfiable concepts in OWL ontologies. In ESWC, volume 6, 170–184. Springer.
- Troquard N, Confalonieri R, Galliani P, Penaloza R, Porello D, Kutz O. Repairing Ontologies via Axiom Weakening. In: Proc. of AAAI 2018; 2018. p. 1981-8.
- Porello D, Troquard N, Penaloza R, Confalonieri R, Galliani P, Kutz O. Two Approaches to Ontology Aggregation Based on Axiom Weakening. In: Proc. of 27th Int. Joint Conf. on Artificial Intelligence and 23rd European Conf. on Artificial Intelligence; 2018. p. 1942-8.

4.2.3 LOT4OCES Methodology

This section describes the workflow of developing an ontology. The workflow follows the Linked Open Terms Methodology, specializing and sometimes customizing the same for the suitability of OCES. While the overall flow and critical steps remain unchanged from the original methodology, this prescriptive standard instils more stringent requirements. In this section, we primarily focus on the first two phases: Ontology Requirement Analysis and Ontology Implementation. The rest of the phases, i.e., ontology publication and maintenance mostly follow the LOT original guideline except the diversion mentioned in R5:.

The ontology development team SHOULD assign people in the following roles listed in the following table as required by the scope of the development.

Role	Responsibility	Required skill
Product owner	To provide a clear vision of the purpose and scope of the ontology, to represent both the customer and end-users, to facilitate communication between the ontology development team and the customer	Decision-making, problem-solving, storytelling, creativity, application knowledge
Domain Expert	To provide general knowledge about the domain of the ontology, to supply the ontology development team with definitions of concepts specific to the target domain or application, and to provide definitions and explanations of concepts from existing standards, corpus, and other ontologies.	Domain knowledge, subject-matter expertise, awareness of relevant standards, corpus, and related ontologies for the target domain
Ontology Analyst	To translate business needs into ontology requirements, to ensure proper documentation of requirements, to define the scope and prioritize development, to verify that requirements are satisfied by proper testing of the ontology	Ontology requirement engineering, critical thinking, negotiation skills, interpersonal skills
Ontologist	To extract concepts from the requirement, to build conceptualization of the ontology, to build definitions of concepts, to ensure that the ontology is semantically meaningful, to help in mapping concepts from different ontologies	Knowledge of different ontological and metaphysical theories, knowledge of different ontological commitments, Expertise in one or more top-level ontologies, and mid-level ontology, expertise in developing taxonomy, partonomy, and ontology
Logician	To write axioms by translating definitions of ontology concepts in some formal logic system, e.g., FOL, OWL 2.0, to ensure logical consistency of the ontology through the use of reasoners and provers, to provide mappings between concepts from different ontologies	Expertise in writing axioms in some logic systems, knowledge of formal logic, and model-theoretic systems, skilled in using provers and reasoners and debugging.
Ontology Developer	To generate the ontology artefact, e.g., OWL/RDF file, to manage import and IRIs, to annotate the ontology, to provide metadata, to generate documentation of the ontology,	Knowledge of ontology languages, encoding formats, proficiency in using ontology editing,

	to manage changes and releases of the ontology artefact.	documentation, and source control tools, knowledge of standard metadata properties
Ontology Tester	To create test cases for the ontology, to write necessary queries or set testing tools for functional testing, to measure the ontology using various metrics, to facilitate the peer-review process, to write test reports	Proficiency in creating test cases, writing queries, and using ontology evaluation tools, knowledge of various ontology metrics and their interpretation
Ontology Reviewer	To review and comment on the model and documentation of the ontology. To evaluate the quality of the ontology for aspects which cannot be automated. To write critiques of definitions and documentation.	Similar skills of ontologist, logician, and domain expert. Multiple reviewers should be recruited from outside the project.
Ontology Curator	To catalogue the ontology in a suitable ontology repository, to generate accurate and sufficient FAIR metadata for the ontology, to measure and improve the FAIRness of the ontology, to manage versions of the ontology	Proficient in using ontology repository, expertise in FAIR metadata management, knowledge of FAIRness evaluation of ontology
Data Modeler	To create data transformation, e.g., ETL, strategy, to create mappings from data sources to ontology, to set knowledge graph database (KGDB) and data pipelines, to validate data represented as KG, create query APIs for the downstream use of data	Proficiency in various data transformation and mapping languages and tools, knowledge in data life cycle management, expertise in data verification and validation, and experience with various KGDB and related ETL tools.

R1.1:

R1.2: The same person MAY be assigned to multiple roles if suitable.

R1.3: For some ontology development projects, the following roles given in the following table MAY be assigned if required.

Role	Responsibility
Ontology Publisher	To endorse and sometimes fund the ontology development, e.g., a standardization body
Ontology Translator	To translate the labels, annotations, metadata, and documentation of an ontology in another language
Project Manager	To manage the ontology development project including organization, planning, and execution, to ensure that the project is completed within the budgets and schedules.

Figure 8 - Optional roles in an ontology development team

R2: The ontology development team SHOULD plan the development lifecycle of an ontology along four broad phases: (1) Ontology requirements specification; (2) Ontology implementation; (3) Ontology publication; and (4) Ontology maintenance, as shown in .

R2.1: Some activities in Ontology implementation and Ontology publication and maintenance phases as a whole MAY NOT be undertaken if the purpose of the project is to use existing ontologies for various data modelling purposes.

R2.1.1: The ontology development team SHOULD undertake activities under Ontology exploitation as shown in Figure 9 of data modelling using existing or developed ontology.

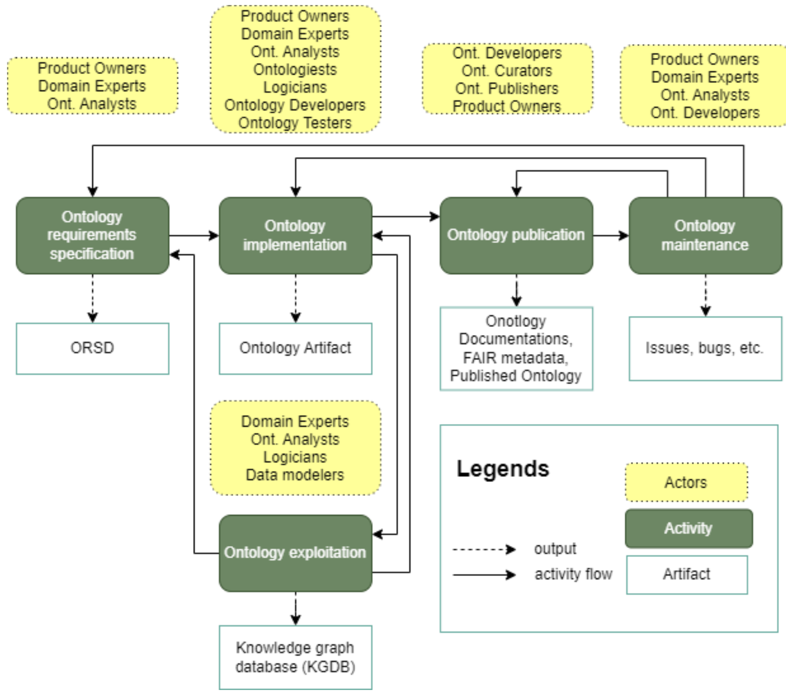

Figure 9 - LOT4OntoCommons base workflow

R3: Whether the purpose is the development of domain or application-level ontology or data modelling using existing or developed ontology, the ontology development team SHOULD undertake activities under the Ontology requirements specification at the beginning of the ontology development project as shown in .

R3.1: The ontology development team SHOULD include members with roles of Product owners, Domain experts, and Ontology analysts for performing activities under Ontology requirements specification.

R3.2: The activities of Ontology requirements specification SHOULD follow the BPMN diagram given in Figure 10.

R3.2.1: The requirements specification SHOULD start from gathering use cases and data sets following the guidelines in R3.3: and R3.5:.

R3.2.2: From the use cases and data sets, the purpose and scope of the ontology SHOULD be identified following the guidelines in R3.6:.

R3.2.3: From the use cases and data sets, a set of user stories is collected as the intended use of the ontology following the guidelines in R3.7:.

R3.2.4: From the user stories, a set of competency questions is developed as the functional requirement for the ontology following the guidelines in R3.8:

R3.2.5: From the competency questions, a set of terms is extracted in a glossary following the guidelines in R3.9:.

R3.2.6: The outputs of the above activities are encoded in the Ontology Requirement Specification Document (ORSD) following the guidelines in R3.10:.

R3.3: The ontology development team SHOULD perform a series of interviews with the product owner and other users to gather Use cases and Data sets.

R3.4: The Ontology analyst SHOULD collect and document use cases in the following template given in Figure 11 by consulting the Product owner. The Ontology analyst MAY also take help from a Domain expert to fill up the template.

R3.4.1: More than one use case specification MAY be created for a project.

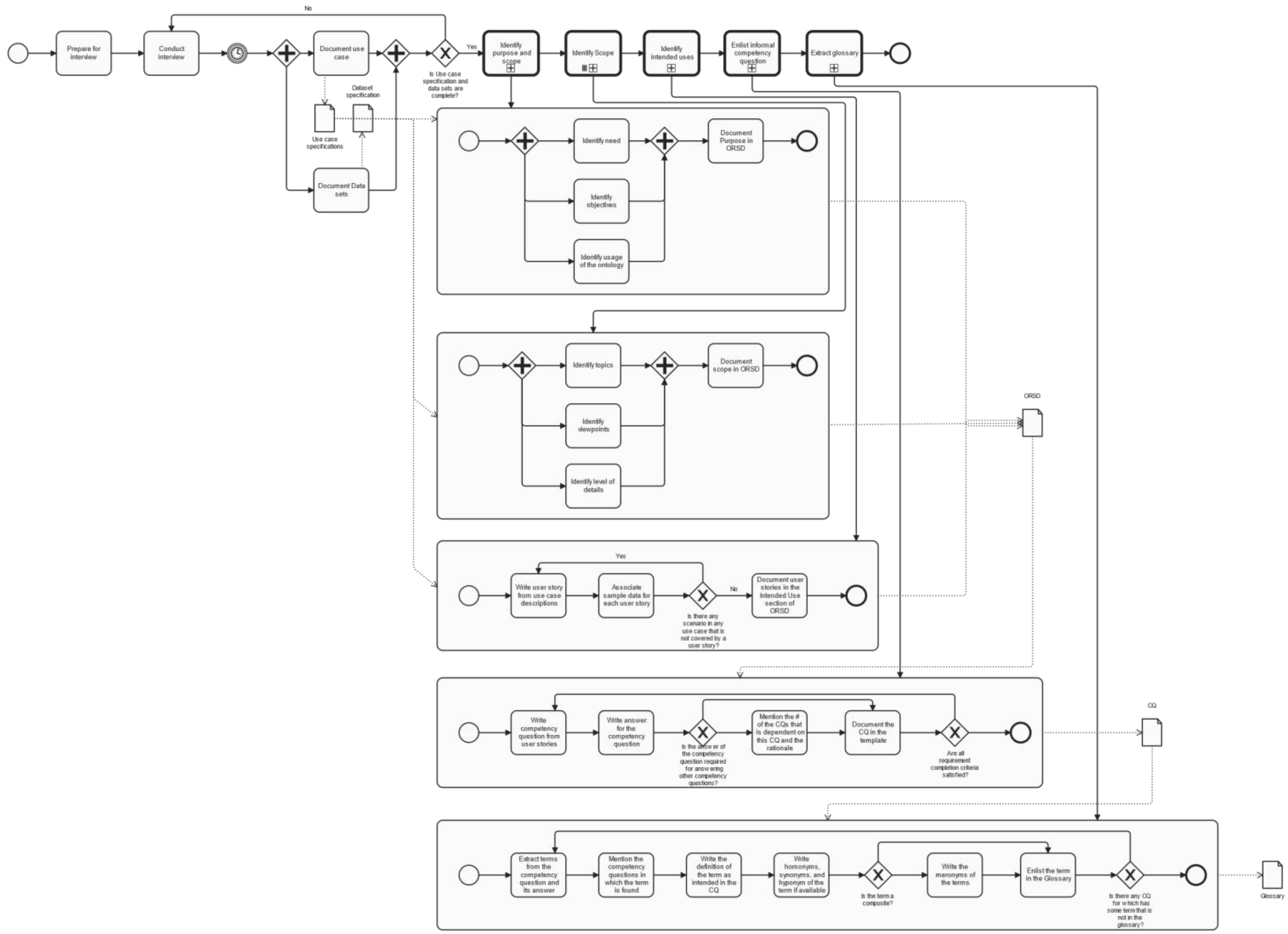

Figure 10 - BPMN diagram for ontology requirement specification phase

<table>
<tr><td colspan="2">Use Case Specification - <identifier/name></td></tr>
<tr><td colspan="2">Introduction: (Short description of the use case)</td></tr>
<tr><td>Use case owner:</td><td>(Short description of the use case)</td></tr>
<tr><td>Involved partners:</td><td>(Product owner / Customer / Client / Vendor / Supplier / knowledge scientist/engineer / material scientist)</td></tr>
<tr><td>Purpose of ontology application</td><td>(Data model / data structuring / Data sharing / Overview and visualization / Context bridging in digital communication / Software customization / Artificial Intelligence / Service extension / Business planning / communication / Decision System / Innovation Project
Workflow / QA/QC / Guided AI / Data Parsing / Data Integration Interoperability / Others)</td></tr>
<tr><td>Data sources used:</td><td>(Mention what data describe; source of data, e.g., database, API, diagram, text; format of data, e.g., csv, tsv, xml, json, text)</td></tr>
<tr><td>Ontologies considered:</td><td>(Mention if any existing ontology is used)</td></tr>
<tr><td>Main Challenges:</td><td>(Main challenges that the clients are facing in the current scenario)</td></tr>
<tr><td colspan="2">Description:
1) Context of the use case: (Describe the context in which the use case takes place, e.g., acquiring leads in a marketing campaign, material handling in an assembly line)

2) What are the pre-conditions of your use case? (Describe the states of the system and environment before the use case scenario starts)

3) What are the post-conditions of your use case? (Describe the states of the system and environment after the use case scenario ends)

4) Who are the actors involved? (List of all actors mentioned in the scenario steps)

5) What are the steps of the main scenario?
(Please add here a picture that describes the aforementioned user story, in a compact way with discrete steps, e.g., workflow. An additional picture in the form of a use case UML diagram with connections to actors and software/hardware components would also be helpful.)

6) Which data sources are used? (Please cross-reference with the steps of the main scenario)</td></tr>
</table>

<table>
<tr><td colspan="4">7) What are the major (expected) contributions of this project to the use case? (Describe how the project's outcome will mitigate the challenges of this use case)

8) Name the most important 20 terms for the domain of your use case. (If available, please add any diagrams illustrating these terms (e.g., UML diagram, ER diagram)</td></tr>
<tr><td colspan="4">Implementation plan:
(Provide a rough time plan for your implementation. Particularly, list the milestones or scenario steps - by when what will be developed. Mention any additional resources that will be available during the project and associated constraints. Include any policy or protocol that the customer wants to follow.)</td></tr>
<tr><td colspan="4">FAIR plan:
1) Is there any existing plan for FAIRness? (Does the customer have an existing FAIRness strategy)

2) What are the future steps to improve FAIRness? (What does the demonstrator want to improve? If there are no/little improvements made and/or foreseen, motivate why. If there is an interest in progress here, what are the roadblocks?)</td></tr>
<tr><td colspan="4">Technology readiness level: (Follow the guides[2,3] for assigning TRLs)
1) of different scenario steps
2) of the application of ontology
3) of different tools involved</td></tr>
<tr><td colspan="4">Key Performance Indicators (KPI): (Provide the KPIs, corresponding metrics, and expected improvement by the end of the project)</td></tr>
<tr><td>KPI</td><td>Metric</td><td>Current</td><td>Future</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

Figure 11 - Use case specification template

R3.5: The Ontology analyst SHOULD collect datasets, regulations, standards, data formats, API specifications, and database schemas from the customer (mediated by the Product owner).

R3.5.1: The Ontology analyst SHOULD document the purpose, type, and origin of the data along with any security concerns.

[2] Mitchell, J.A., Measuring the Maturity of a Technology: Guidance on Assigning a TRL, SAND2007-6733, Sandia National Laboratories, Albuquerque, NM, October 2007.
[3] https://en.wikipedia.org/wiki/Technology_readiness_level.

R3.5.2: The Ontology analyst SHOULD mention the related use case.
R3.5.3: The Ontology analyst SHOULD document how the data can be accessed.
R3.5.4: The Ontology analyst SHOULD document how long the data will be available when it will be updated and any other restrictions on its use.
R3.5.5: Ontology analysts SHOULD analyze the data for extracting potential features and the relationships among them in consultation with the domain expert and product owner.

3.5.5.1. For a set of APIs, the JSON data model MAY be documented. In this analysis, the API specification document and sample of API requests and responses need to be documented.

3.5.5.2. For relational data in database tables, the schema for tables, fields, relationships and viewed MAY be documented along with constraints using both primary and foreign keys.

3.5.5.3. For comma-separated, tab-separated files, and spreadsheets (common for logs, and records) the fields MAY be documented as well and more persistent storage of the record, possibly in some databases MAY be investigated for more information.

3.5.5.4. For unstructured data, e.g., free text, images, and audio, common metadata for the data files MAY be collected. Exploring the unstructured data requires further processing. These techniques are out of the purview of this guide.

R3.5.6: Every set of data should be documented by using the template in Figure 12.

DATA SET/Data Exchange Format n. # – <title of the data set> Owner(s): <organisation>		
1. DATA SUMMARY	Purpose of the Data*	
	Type and Format of data*	
	Reused-Data	
	Data origin	
	Data size / Data exchange frequency	
	Data security and storage	
	Discoverability of data or API specification (metadata)	
	Use case #	

2. DATA STRUCTURE	Name of columns or fields or objects	Name | Description
	Standard vocabulary or mapping to commonly used ontologies	
	Identifiability of data/transaction or primary and foreign key	
	Naming conventions used	
	Data model/schema/entity-relation diagram	
3. DATA ACCESS	Database/repository/folder path/URI	
	Sample queries	
	Methods or SW tools for data access	
	SW documentation and other information needed	
	Join conditions, filters, aggregation	
	Search keywords approach	
4. DATA AVAILABILITY	Timing of data availability (incl. indications on embargo)	
	Data update cycle	
	Data usability by Third Parties (after the end of the project)	
	Restrictions to data use/Access control	
	Expiration date	
5. DATA OWNERSHIP	Data owner	
	Data manager	

Figure 12 - DATA SET/Data Exchange Format Form

R3.6: The purpose and scope identification activity task MUST be carried out by domain experts, product owners, and ontology analysts.

R3.6.1: The purpose and scope MUST be identified by analyzing the use cases and the domain documentation provided in the data exchange identification.

R3.6.2: The purpose statements SHOULD provide a set of "completion criteria" for the ontology engineering project. The statements in the purpose SHOULD include the following aspects:

a) Statements on the needs of the project, e.g., statement of need identifying the source of the request (person or project) and paraphrasing the stated motivations for the project.

b) Statements on the objectives that the output of the project should satisfy, e.g., decision support system, integration of system, query answering system, improvement of reliability and quality, communication among departments, entities, and systems, knowledge acquisition, automation, classification, detection, and prediction. The statement of objectives should be "SMART", i.e., specific, measurable, achievable, relevant, and time-bound.

c) Statements on the usage of ontology in data management tasks, e.g., annotation, configuration, filtering, indexing, integration, matching, mediation, personalization, query formulation, query rewriting, and search.

R3.6.3: The objectives included in the purpose SHOULD be quantified in terms of KPIs.

R3.6.4: The scope statements SHOULD define the context of the project in terms of:

a) The topics which the ontology covers and which it should not, e.g., domain or application area, phases of the lifecycle, types of activities, types of resources, part of the system, list of protocols, genre, practice, paradigm, and types of documents.

b) The viewpoint the ontology commits to, e.g., the perspective of the user, frame of reference, and mission and vision of the client's organization.

c) The level of detail for the ontology development effort, e.g., existing ontology to be exploited, existing standards to follow, various format-related requirements (IRI format, encoding format etc.), level of expressivity, and intended end users.

R3.6.5: The scope statements MAY be updated at a later stage of the project by the assessment of the ontology analyst in agreement with the product owner.

R3.6.6: The scope analysis MAY bring out several non-functional requirements, especially from R3.6.4:c).

R3.7: A set of user stories MUST be extracted from the use cases and datasets by ontology analysts while being supported by domain experts, and the product owner.

R3.7.1: For each user story, a sample of data MAY be associated if required.

R3.7.2: Each user story SHOULD be identified with a numbering scheme.

R3.8: The Functional Ontology Requirement Proposal MUST be prepared by ontology analysts while being supported by domain experts, and the product owner.

Definitions
Competency Questions: Given a use case, the set of queries which place demands on an underlying ontology for a certain level of expressiveness such that the ontology must be able to represent these questions using its terminology and be able to characterize the answers to these questions using the axioms and definitions, is called competency questions. **Informal Competency Questions**: The competency questions that are not yet expressed in the formal language of the ontology are called informal competency questions[4]. **Formal Competency Questions:** The competency questions that are defined formally as an entailment or consistency problem with respect to the axioms in the ontology.

R3.8.1: Functional Ontology Requirements SHOULD be materialized as Competency Questions.

R3.8.2: Informal Competency questions MAY be derived from user stories extracted from use cases as mentioned in R3.7:.

R3.8.3: Informal Competency Questions MUST be written by the ontology analyst with the help of the domain expert.

R3.8.4: The competency questions SHOULD be defined in a stratified manner, with higher-level questions requiring the solution of lower-level questions[5].

R3.8.5: For any competency question, the rationale MAY be specified for the question (which states how the answer to the question is used to answer more complex questions) and/or specify the decomposition of the question (simpler questions which we must answer to answer the given question).

R3.8.6: Every proposal for a new or extension of the existing purpose and scope and associated amendment to the use case SHOULD be covered by a set of additional competency questions.

R3.8.7: The template given in *Figure 13* - Functional requirement specification template MAY be used to write and manage informal competency questions.

---

[4] By specifying the relationship between the informal competency questions and the motivating scenario, we give an informal justification for the new or extended ontology in terms of these questions. This also provides an initial evaluation of the new or extended ontology; the evaluation must determine whether the proposed extension is required or whether the competency questions can already be answered by existing ontologies.

[5] It is not a well-designed ontology if all competency questions have the form of simple lookup queries; there should be questions that use the solutions to such simple queries.

R3.8.8: The set of competency questions MUST be checked for their validity and completion by the product owner and ontology analysts.

3.8.8.1. The set of completion criteria given below SHOULD be used for testing the validity and completion of competency questions.

- A set of competency questions is correct if each competency question is about some aspect of a user story that needs to be modelled by the ontology.
- A set of competency questions is complete if no additional competency question is required to be added to the set, i.e., there are no user stories which has some aspect that is not covered by the competency questions.
- A set of competency questions is internally consistent if no conflicts exist between them, i.e., for every competency question, the assumption of correctness of its answer should not imply the answer to any other competency question is incorrect.
- A set of competency questions is verifiable if there is a finite process with a reasonable cost that tests whether the final ontology satisfies each requirement, i.e., there should not be any competency question which requires access to additional data (not associated with the user story the competency question is about), manual interventions, or execution of algorithms to derive the answer.
- A set of competency questions is comprehensible if every competency question is understandable to the product owner and the ontology analyst.
- A set of competency questions is unambiguous if every competency question has only one possible answer for every situation.
- A set of competency questions is concise if every competency question is relevant, i.e., no duplicated or irrelevant competency question exists in the set.

3.8.8.2. The ontology analyst MAY declare the completion of the requirement analysis and signal the start of the implementation phase only when the set of competency questions passes the validity and completion test.

Identifier	Sprint	Competency Question	Answer	Status	Higher level Question	Rationale	Comments	Extracted from (provenance)	Priority
Code (domain+id)	Code or label the sprint	An interrogative sentence	Number, Token, sub-sentences, or a complete sentence	Proposed, Accepted, Rejected, Pending, Deprecated	Code	Text	Text	Source of the collected CQ	High, medium, or low

Figure 13 - Functional requirement specification template

Identifier: The requirement identifier needs to be unique for each requirement. A suitable naming convention needs to be decided by the project members, e.g., use case # + index, domain + index, application + index.

Sprint: The sprint of the requirement iteration. Multiple iterations may be undertaken at the beginning of the project. Separate sprints needs to be undertaken for new amendment to the purpose and scope of the use case as well as an update of the ontology.

Competency Question: The competency question.

Answer: The answer to the competency question.

Status: The status of the requirement might be: 1) proposed, 2) accepted, 3) rejected, 4) ongoing or 5) deprecated.

Higher Level Question: The # of competency questions that depend on the answer to this competency question to be answered.

Rationale: How the answer to the question is used to answer the higher-level question, e.g., provides constraints (structural, functional, quantitative, qualitative, spatial, temporal, causal etc.), provides data (create new instance, asserts new relation, record measurement etc.), provides assumptions (create new situation, initiation of process, new activity etc.).

Comment: Free text for keeping notes regarding the competency question.

Provenance: The provenance of the provided requirement (use case, interview, mail, etc.).

Priority: The priority of the requirement can be: 1) high, 2) medium, or 3) low.

R3.9: A list of terms MUST be identified from the informal competency questions in a glossary.

R3.9.1: The glossary MUST be built by the Ontology Analyst.

R3.9.2: The keywords (e.g., subject, object, and main verb) for every informal competency question and its corresponding answer (e.g., tokens) SHOULD be enlisted in the terminology.

R3.9.3: The competency questions from which a term is extracted MUST be indicated for that term.

R3.9.4: Every term SHOULD be explained by associating the intended meaning of the term in the corresponding competency question.

R3.9.5: Every term MAY be associated with other homonyms, synonyms, hyponyms, and meronyms from a reputed source, e.g., Wordnet, dictionary etc.

- Synonyms could be found for most of the terms while only some terms have homonyms.
- If the term is a composite then there exists some meronyms.

R3.9.6: The template given in **Error! Reference source not found.** may be used to enlist the t erms in the glossary.

#	Term	Definition	CQ Identifiers (comma-separated)	Synonym	Homonym	Hyponym	Meronym

Figure 14 - Glossary Template

#: Index of the terms.

Term: Single word or a phrase

Definition: The definition of the word that corresponds to the meaning in which the term is used in the corresponding competency questions. If the same term is used with different meanings in some competency questions then they should be enlisted with different indexes.

CQ identifiers: The list of CQs from which the term is extracted (separated by a comma).

Synonym: The words or phrases that mean exactly or nearly the same as the term (separated by a comma).

Homonym: Other meanings of the term (separated by a comma).

Hyponym: The words or phrases with broader meaning (superordinate or supertype) of the term if available.

Meronym: The words or phrases which correspond to the part of the entity defined by the term if available.

R3.9.7: The terms MAY be sorted according to their frequency of occurrence in the competency questions.

R3.9.8: The competency questions MAY be prioritized based on the frequency of the terms calculated in R3.9.7:.

R3.10: Once the ontology development team has all the information about the requirements, obtained from the activities described in R3.6:, R3.7:, R3.8:, R3.9:, and the Ontology Requirements Specification Document (ORSD) MUST be created following the template given in **Error! Reference source not found.**.

R3.10.1: Only one ORSD SHOULD be produced for each sprint of the project.

R3.10.2: An ORSD MUST contain "Name of the project", "Prepared by" (name of the person who signs off the ORSD), "Purpose", "Scope", "Functional Requirement", and "Glossary of Terms".

3.10.2.1. The output of R3.6.2: and R3.6.3: SHOULD go to the field "Purpose".

3.10.2.2. The output of R3.6.4: SHOULD go to the field "Scope".

3.10.2.3. The output of R3.7: SHOULD go to the field "Intended Uses".

3.10.2.4. The Functional Requirement table produced by the guideline of R3.8: SHOULD either be embedded or hyperlinked in the field "Functional Requirements".

3.10.2.5. The Glossary table produced by the guideline of R3.9: SHOULD either be embedded or hyperlinked in the field "Glossary of Terms".

R3.10.3: An ORSD MAY contain "Non-Functional Requirements".

3.10.3.1. The output of R3.6.6: SHOULD go to the field "Non-Functional Requirements".

Ontology Requirements Specification Document	
1	Name of the project and sprint
2	Prepared by
3	Purpose
4	Scope
5	Intended Uses
User story # 1 … User story # 2	

…	
6	Ontology Requirements
	1. Non-Functional Requirements
	2. Functional Requirements
	Link to Functional Requirement Specification
7	Glossary of Terms
	Link to the glossary

Figure 15 - Ontology requirement specification template

R4: After the Ontology Requirements Specification phase has produced the ORSD completely and especially the functional requirements have passed the completion criteria mentioned in R3.8.8:, the ontology development team SHOULD undertake activities under the Ontology Implementation as shown in Figure 1.

R4.1: The ontology development team SHOULD include members with roles of ontologists, logician, ontology analysts, domain expert, ontology developer, and ontology tester for performing activities under Ontology Implementation.

R4.2: The activities of Ontology Implementation SHOULD follow the BPMN diagram given in Figure 16 - Ontology Implementation workflow

R4.3: .

R4.3.1: The Ontology Implementation phase SHOULD be performed iteratively by taking some terms from the glossary into account in each sprint. The ontology analysist SHOULD schedule and plan the ontology development by dividing the glossary into smaller subsets and prioritizing them according to the number of occurrences of CQs for each term (see R3.9.3:).

R4.3.2: The Ontology Implementation MUST start with the parallel activities of ontology conceptualization and reuse ontology following the guidelines in R4.3, which produces a set of concept analysis documents.

R4.3.3: Based on the information on concept analysis documents, a set of decisions SHOULD be made as guided by R4.3 on whether ontology encoding SHOULD be performed or to move to other sub-activities that are shown in Figure 16 - Ontology Implementation workflow

R4.3.4: .

R4.3.5: Based on the decision made in the step R4.3.3:, activities under ontology encoding MAY be performed as guided by R4.3.6 to produce ontology source file(s).

R4.3.6: Encoded ontology source file(s) MUST be tested by performing activities under ontology evaluation.

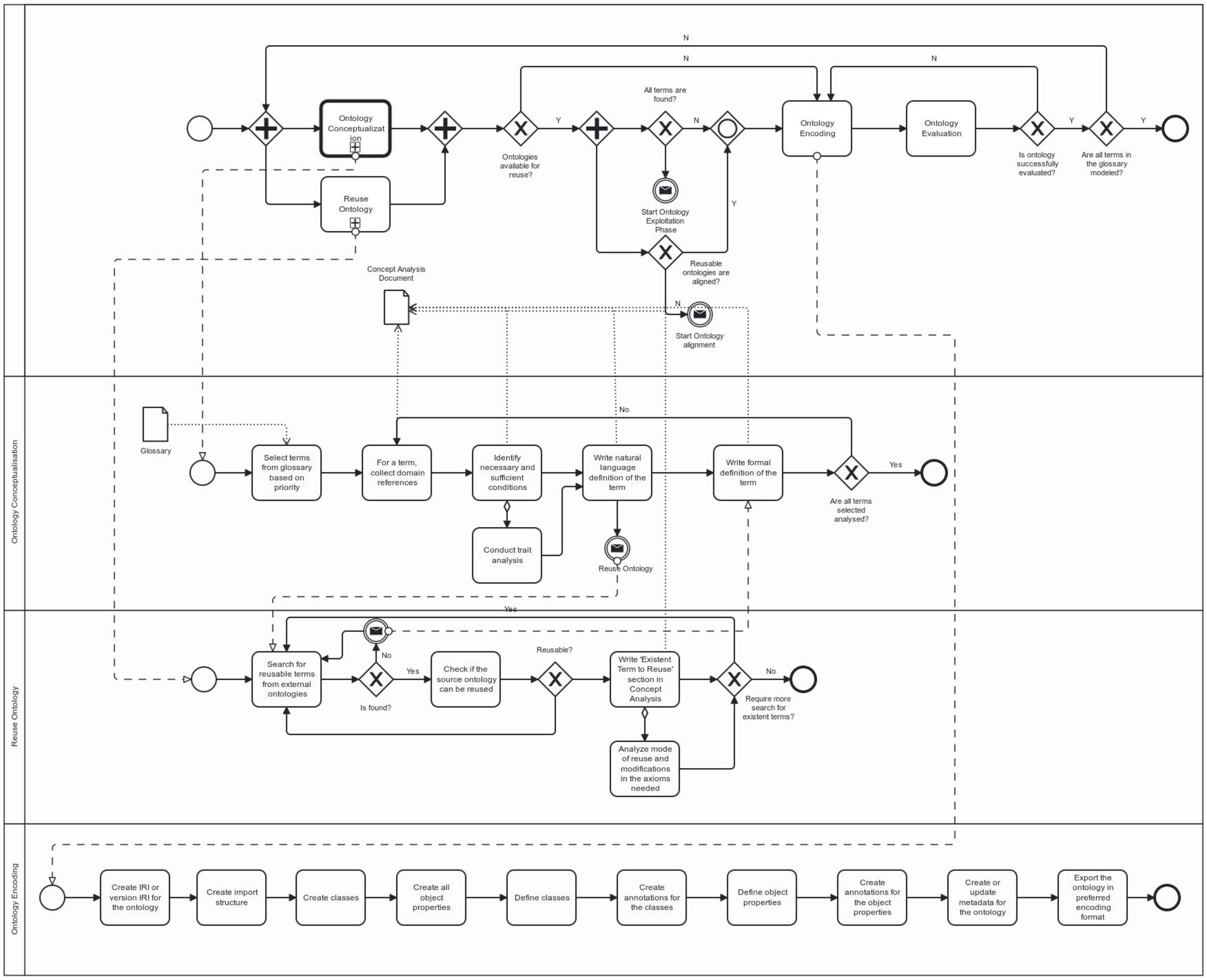

Figure 16 - Ontology Implementation workflow

R4.4: The ontology development team SHOULD engage members with roles of Ontologist, logician, domain expert and ontology analyst to perform ontology conceptualization.

R4.4.1: Each term considered for modeling in the current sprint SHOULD be analyzed for the concept it represents to ultimately find a suitable definition. The template given in **Error! Reference source not found.** MAY be used to capture different aspects of t he analysis as guided below.

4.4.1.1. Various labels that MAY be associated with the concept is enlisted in the 'Labels' section of the 'Knowledge Domain Resources' in **Error! Reference s ource not found.** by ontology analysts and domain experts. The labels may be classified into preferred labels, alternative labels, and hidden labels.

4.4.1.2. Domain experts SHOULD collect existing domain resources (e.g., standards, books, articles, dictionaries) that define or are related to the concept with reference to the source in the section 'Related domain resources' of the 'Knowledge Domain Resources' in **Error! Reference source not found.**. M ore than one resource can be reported.

4.4.1.3. As the concept may or may not completely align with the elucidations from the domain resources, ontology analyst and domain expert SHOULD collaborate to delineate what clauses and conditions are to be included in the definition of the concept in the context of the requirement. Any associated comment MAY be documented in the 'Comments' section of the 'Knowledge Domain Resources' in **Error! Reference source not found.**.

4.4.1.4. The definition of the term MUST be written in the 'Definition' section of the 'Knowledge Domain Resources' in **Error! Reference source not found.** to e lucidate the concept the term represents, following the best practices of definition writing. The information found in the domain in 'Related domain resources' and the 'Trait Analysis' MAY be considered in formulating the definition of the concept.

4.4.1.4.1. 'Trait analysis' section of the 'Knowledge Domain Resources' in **Error! Reference source not found.**, MAY be used to capture t he analysis conducted following the best practices of definition writing.

4.4.1.5. Before the 'Formal Definition' of the term may be encoded, activities under Reusing Ontology MUST be performed as guided by R4.4.2:. During these activities, the 'Existent Terms to Reuse' section MAY be filled by the steps described in 4.4.2.3 and 4.4.2.4.

NEW TERM NAME

(use the preferred label, or IRI name, to be used as the file name)

General Concept Info:

IRI:	**Suggested entity new IRI.**
OWL Type:	*Class|ObjectProperty|Individual*
Natural Language Definition:	*Natural language definition of the concept (elucidation).* *Here the concept that we want to introduce is expressed as precisely as possible, making references to knowledge domain resources, including instance and usage examples when relevant. The necessary and sufficient*

	conditions for the definition can be found by applying trait analysis. Follow the 'best practice for definition' while writing the definition.
Formal Definition	*Definition written in the formal language (FOL, DL etc.)*
Labels:	*Labels used to address the concept, ordered as:* *i) preferred (one) (the label to primarily used to shortly refer to the concept)* *ii) alternative (multiple) (labels that are commonly used to address the concept in practice, even if they are used with narrower of wider sense)* *iii) Hidden (multiple) (labels that should be hidden when generating visual displays of the resource, but should still be accessible to free text search operations.).*

Knowledge Domain Resources:

Related Domain Resources:	*Existing domain resources (e.g., standards, books, articles, dictionaries) that define or are related to the concept (provide reference to the resource and quote the relevant informational content).* *More than one resource can be reported.* *These resources are aimed to support the choice of the above concept choice and elucidation.*
Comments:	*Explain the motivations behind the concept definition with reference to the domain resources, underlying similarities and differences.*

Trait analysis:

Traits	*List of all possible traits for the term*
Analysis	*Trait analysis table and rationales* Necessary conditions: Sufficient conditions:

Existent Terms to Reuse:

IRI:	*Entity IRI*
OWL Type:	*Class/ObjectProperty/Individual*

Source Ontology	*IRI, prefix and title of the source ontology*
Concept Elucidation	*Refer to ontology annotations or other existing official documentation that defines the concept in natural language.* *Here the axioms that acts as definition or constraint for the term may also be documented.*
Comments	
Mode of Reuse:	*As-is / Extracted / Extended / Combined*
Axioms to be Modified:	*Lists axioms that are removed from the existent ontology or added in the new ontology for the purpose of reusing the concept.* *In case Combined with other terms from different ontologies, the terms should be prefixed and documented in separate block under Existent terms to reuse.*

Figure 17 - Concept Analysis Template

R4.4.2: After the terms are clearly defined, existing ontologies MUST be searched for suitable concepts that can be reused either as is or as part of the definition as part of Reuse Ontology phase. The following are the directions for reusing ontology.

4.4.2.1. Domain experts, ontologists and logicians SHOULD collaborate to identify existing ontologies that may contain terms for reuse by searching different ontology repositories and registries or from their own experience. The search of reusable terms MAY also takes help of ontology alignment tools. Some ontology portals may provide tools, e.g., search, recommendation, annotation.

4.4.2.2. In reusing ontology, the priority should be given to the ontologies in the following order: 1) ontologies which are aligned to a top-level ontology, 2) ontologies which are aligned to some ontologies in the OntoCommons ontology stack, 3) ontologies that are being used by the customers if exists, 4) ontologies which have been standardized (e.g., by ISO, W3C, standardization focused on domain etc.) or developed by some reputed organizations through collaborative efforts, 5) ontologies that are popular in the particular domain or application, 6) ontologies recommended by the ontology development team or customers.

4.4.2.3. The reusable ontology MUST follow the definition 2 of [1]. If accepted, the term should be documented in the 'Existent terms to reuse' section of the template in **Error! Reference source not found.** by providing the IRI, prefix a nd title of the source ontology, the label of the term, its type, e.g., class, relation or individual, and the definitions of the term providing the elucidation of the concept the term represents. For each term that is a candidate for reuse from same or different ontology MUST be documented in a separate section in the 'Existent terms to reuse' section.

4.4.2.4. The term which is deemed to be reusable SHOULD be analyzed by the ontologist and logician to compare the theory supporting the term fits to the intended model of the ontology being developed and to estimate what sort of changes may be required to be made to the original axioms for fitting candidate term for reuse can be fitted to the overall model of the ontology being developed. Following the types of modification operations mentioned in [1], an appropriate set of modification operations also SHOULD be recorded in the 'Modes of reuse' section of the template in **Error! Reference source not found.**. Additionally, the axioms to be r emoved from the source ontology in case of extraction and the axioms to be added in case of extension and combination MAY also be recorded. However, it may be difficult to precisely formulate such modifications until all terms are analyzed and the set of ontologies to be reused is determined. For this reason, they may be identified at the later stage of Ontology Conceptualization or during Ontology implementation.

4.4.2.5. Apart from the directives in 4.4.2.2, 4.4.2.3, and 4.4.2.4, the decision to reuse ontology SHOULD consider the following the following aspects.

4.4.2.5.1. Ontologies at different levels requires to reuse different types of ontologies. A mid-level ontology MUST reuse a suitable top-level ontology and some other mid-level ontology to reuse. In these cases, the practice is to find one top-level ontology to follow but the development of alignments among different top-level ontologies as part of OntoCommons Ecosystem MAY provide opportunity to use multiple top-level ontologies. A domain reference or domain reference ontology on the other hand MAY reuse multiple mid-level ontologies as well as other domain ontologies. An application-level ontologies MAY reuse multiple domain level ontologies. In all cases, the ontology team SHOULD give priority to reuse ontologies which are aligned to the same top-level ontologies or multiple top-level ontologies which are aligned among themselves.

4.4.2.5.2. The terms being reused from more than one ontology MUST be semantically consistent. If the terms are formally well-defined (e.g., classes carry necessary and sufficient conditions and properties have domain and ranges and implications clearly encoded), then the reasoners may be used to check consistency. Furthermore, being aligned to a top-level ontology is often helpful for understanding the correct interpretation of the terms. However, external ontologies may not contain enough axioms to capture all the constraints intended by their developers. In these cases, further investigations by studying the annotations, documentations, and use cases of these external ontologies MAY required to be performed to find the interpretations of the classes and constraints on the relations. It is recommended that in case of ambiguity or insufficient information, the ontologists SHOULD NOT impose their own interpretation on an external term.

4.4.2.5.3. Reusing ontologies always comes with extra cost in terms of development and maintenance. The cost increases with the number of ontologies being reused as well as how the terms from these ontologies are being reused. Considering this, the number of reused ontologies SHOULD be kept to a minimum as much as possible.

R4.4.3: If no concept analysis included mapping to existing term, then the development team MAY start encoding the ontology without reusing to any existing ontology. If at least one existing concept is reused, then proceed to the check described in R4.4.4:. It is however extremely unlikely that the no ontology can be found for reuse, including even some top-level ontology for them being domain-neutral. Although, LOT4OCES does not prohibit developing an ontology without mapping to existing ontologies, the ontology development team MUST make every effort to find suitable ontologies to reuse.

R4.4.4: If the existing terms reused have different source ontologies, then it MUST be checked whether the source ontologies are aligned among themselves.

4.4.4.1. If at least one of the source ontology is not aligned to the rest of the source ontologies, then ontology alignment process MUST be undertaken by following the guideline in Sections **Error! Reference source not found.** and **Error! Reference source not found.** before proceeding to the next step.

R4.4.5: If every concept has an equivalent existing term that can be reused from an existing ontology, then no further ontology development is needed and the development team MAY proceed to ontology exploitation activities, e.g., mapping data to ontology, annotating data sources, and use for ontology exchange. If only a subset of the terms from the glossary is considered for development, then this decision MUST be taken after all terms are analyzed.

R4.4.6: After all terms, considered for the current sprint, are completely analyzed, and went through the checks in R4.4.3:R4.4.4:, R4.4.5:, the ontology should be encoded by following the guideline below by the ontology developer, with support from the logician.

4.4.6.1. The IRI or the version IRI MUST be determined by following the IRI convention in <>.

4.4.6.2. The import structure of the ontology SHOULD be configured following the OCES ontology stack in Section 4.2.2. It is highly recommended that reused ontologies are imported in the ontology file. However, sometimes it may be preferred to not insert the IRIs of the imported ontologies directly in the file but maintained outside of the ontology source. This decision SHOULD be taken considering the infrastructure of maintaining ontology at the customer's side.

4.4.6.3. The class terms MUST be created by selecting either opaque or transparent IRI scheme (see Section 5.1). For every class terms all labels from the 'Labels' section of the analysis document MUST be added. Subclass relations or other axioms MAY NOT be added to the classes at this point.

4.4.6.4. The object property terms MUST be added created by selecting either opaque or transparent IRI scheme (see Section 5.1). For every object property, all labels from the 'Labels' section of the term analysis document MUST be added. Sub-property relations or domain and range constraints MAY NOT be added to the classes at this point.

4.4.6.5. The necessary and sufficient conditions of each class MUST be encoded following the 'Formal Definition' in the corresponding concept analysis, using the following patterns. The patterns are given in TURTLE but any other encoding format or an editor specific mechanism MAY be adopted.

4.4.6.5.1. A necessary condition MUST be written in the following pattern: [Class IRI] rdfs:subClassOf [A different Class IRI from same ontology / a class IRI from a reused ontology / a class expression]

4.4.6.5.2. A sufficient condition MUST be written in the following pattern: [A different Class IRI from same ontology / a class IRI from a reused ontology / a class expression] rdfs:subClassOf [Class IRI]

4.4.6.5.3. A necessary and sufficient condition MUST be written in the following pattern: [Class IRI] owl:equivalentTo [A different Class IRI from same ontology / a class IRI from a reused ontology / a class expression]

A [class expression] MAY be an intersection or union or restriction statement using some Class IRIs from same ontology except the class being defined or a class IRI from a reused ontology.

4.4.6.6. The sub-property relations as well as the domain and range restrictions MAY be added to every object property following the 'Formal Definition' in the corresponding concept analysis. Every object properties MAY NOT have domain and range restriction. Most of the object properties at the domain ontology SHOULD be sub-property of some upper-level property from a reused ontology. Sub-properties inherit the domain and range restrictions of the parent relations.

4.4.6.7. The necessary data-properties MUST be created along with their labels and sub-property, domain and range restrictions. Most of the data properties SHOULD have domain and range restriction. The range restriction of a data property is some xsd datatype. Sub-properties inherit the domain and range restrictions of the parent relations.

4.4.6.8. Every class, object property and data property MUST be annotated by following the annotation guide given in Section 5.2. Every term SHOULD be annotated following a consistent set of vocabularies selected by the ontology developers in collaboration with the ontology analyst from the annotation guide given in Section 5.2.

4.4.6.9. The ontology source MUST be annotated by following the metadata recommendation.

4.4.6.10. The ontology source file MUST be exported in a preferred encoding format.

R4.4.7: The ontology MUST be evaluated by the ontology tester, ontologist, logician, domain expert, ontology analyst, and peer reviewer as guided below.

4.4.7.1. The ontology MUST be tested for its structural consistency considering only its structure, including the logical consistency and general coherence.

4.4.7.1.1. The ontology MUST be tested by a suitable reasoner or prover depending on the language and expressivity used to build the ontology.

4.4.7.1.2. The performance of the A-box reasoning or question answering for the ontology also MAY be tested by a suitable reasoner.

4.4.7.1.3. The ontology SHOULD be checked for patterns in the structure which are problematic in terms of best practices and stability in the future. Some of this problems MAY be detected by checking the ontology using Ontology Pitfall Scanner![6] Some of the pitfalls that are common to occur are:

- Missing inverse object properties.
- Missing domain and range in the object properties.

6

- Creating unconnected ontology elements.
- Missing disjointness
- Using different naming criteria for terms
- Using recursive definitions.

4.4.7.1.4. The terms of the ontology MUST have the minimum annotations as defined in 5.2.

4.4.7.1.5. The terms SHOULD be understandable from the logical definitions and the associated annotations by the domain experts.

4.4.7.2. The ontology MUST be evaluated for its accuracy in satisfying the functional requirements.

1.1.1.1.1. The functional testing MUST be tested against competency questions.

- The competency questions gathered in R3.8: SHOULD be rewritten using the terms of the ontology to facilitate writing formal queries.
- The competency questions MUST be tested against both positive and negative test A-box created from the portion of the data gathered in R3.4:.
- The accuracy of the ontology MUST be calculated based on the number of an information retrievals (true positives, false positives, and false negatives) to derive the metrics precision, recall, and the F-measure (harmonic mean of precision and recall).

1.1.1.1.2. The functional evaluations SHOULD also include the computation of the coverage of the ontology against the glossary.

4.4.7.2.1. The functional testing MAY apply black box methods which includes user agreement: after all conceptualization is in the user's mind. debate, trials, and consensus reaching methodology, and use satisfaction assessment: multiple users, survey, and popularity.

4.4.7.3. The ontology MUST be peer-reviewed by and external reviewer for the following aspects:

4.4.7.3.1. The ontology SHOULD follow a top-level ontology.

4.4.7.3.2. The ontology SHOULD classify terms correctly under the top-level ontology.

4.4.7.3.3. The class hierarchy of the ontology MAY satisfy the OntoClean rules.

4.4.7.3.4. The definitions of the terms of the ontology SHOULD NOT be too restrictive to allow re-usability. For example, judicious use of covering axioms, universal quantifiers, domain and range restrictions for object properties.

4.4.7.3.5. The necessary and sufficient conditions for the terms SHOULD NOT be too broad to avoid misinterpretation.

4.4.7.3.6. The ontology conceptualization SHOULD have minimum ontological bias that they be specified at the knowledge level without depending on a particular symbol-level encoding. An encoding bias results when representation choices are made purely for the convenience of notation or implementation.

4.4.7.3.7. The ontology conceptualization SHOULD have minimal ontological commitment that the ontology should require the minimal ontological commitment sufficient to support the intended model. The ontology should make as few claims as possible about the world being modelled, allowing the re-usability and flexibility.

4.4.7.3.8. The ontology SHOULD be parsimonious in creating object and data properties. Property-heavy semantics should be avoided, instead semantics should associated more with the classes.

4.4.7.4. The ontology evaluation MAY use the following template as a guide to conduct evaluation in a proper way.

		Introduction	
#	Source	Requirement	Description
In1		Title	The title of the ontology.
In2		Location	Where can the ontology (currently) be accessed.
In3		Purpose	Provide an overview of the purpose of the ontology in the context of the company's data and knowledge management.
In4		Ontology generality	Is the ontology an upper-level ontology, a middle-level or domain ontology, or an application/task ontology?

		Functional requirements	
#	Source	Requirement	Description

Fu1		High-level requirements	Is the ontology linked to a set of high-level requirements (for instance, business/domain experts' requirements or research questions)?
Fu2*			If so, does the ontology satisfy (all of) them?
Fu3		Competency questions	Is the ontology linked to a set of competency questions?
Fu4*			If so, how are the competency questions serialized (e.g. natural language, SPARQL, DL query)?
Fu5*			Is reasoning used to answer the competency questions (e.g., if R is a transitive relation, and the facts R(a,b) and R(b,c) are stated, when querying for x such that R(a,x), is c also returned)? Why yes or why not?
Fu6*			Were the competency questions executed against an ontology containing data?
Fu7*			If so, is there the expectation that there will be scalability issues when the queries will be run against data when in production?

		Structural/topologic testing	
#	Source	Requirement	Description
S1		Metric testing	Has the ontology been tested using a tool which extracts quantitative metrics about the ontology structure?
S2*			If so, are there any values that could indicate bad/good quality (for each, if any, indicate way it is so)?
S3		Structural good-practices	If the ontology is in the RDF language, what was the evaluation result using OOPS?

Logical, ontological, and terminological aspects			
#	Source	Requirement	Description
On1		Logical properties	Which language is the ontology expressed in?
On2			Is this a good language for the goal of the ontology (e.g., is the open/closed-world assumption reasonable if employed?
On3			Does this language entail that some intended models are excluded from the ontology, or that some unnatural constructs have to be used?
On4			Is the ontology consistent?
On5			Does the ontology support particular reasoning tasks?
On6		Ontological properties	Do the axioms present in the ontology clearly and correctly model the target domain?
On7			Is the ontology aligned with an upper ontology?
On8			If not, why, and has the ontology's taxonomy been analysed using e.g. OntoClean or other Applied-Ontology-methodologies?
On9			Is the ontology aligned with some middle/domain-level or application/task-level ontology?
On10			Are there any (additional) pre-existing middle/domain-level or application/task-level ontologies that could have been reused?
On11		Concept coverage	Does the ontology cover the relevant concept of the domain?

On12			In which way was it tested?
On13			Does the ontology conform to, or is linked to, some pre-existing standards (if not explain why)?
On14		Terminology	Does the ontology conform to some guidelines for annotations?
On15			Are common terminological and naming conventions respected?
On16			Are the annotations clear and satisfactory for users?
On17			Is there any documentation describing the ontology?

		Expert' and users' feedback	
#	Source	Requirement	Description
Ex1		Users	Has the ontology been evaluated by some ontology-experts?
Ex2			Has the ontology been evaluated by some domain-experts?
Ex3			Has the (application/task build using the) ontology been evaluated by prospective users?

		FAIRness requirements	
#	Source	Requirement	Description
Fa1		FAIRness	Has the ontology been evaluated with respect to FAIRness?

Fa2*			If so, how and with which result?
Fa3			In particular, is the ontology openly available?
Fa4*			If not, for what reason and could, at least the schema or a module of the schema being made available?
Fa5			Where is the ontology hosted and how will it be able to be accessed long term?

Ontology lifecycle requirements			
#	Source	Requirement	Description
Li1		Ontology lifecycle	Is the ontology expected to evolve in the future?
Li2			Is there some staff in charge of maintaining and/or updating the ontology?

Conclusion			
#	Source	Requirement	Description
Co1		Suggestions	Based on the evaluation findings, suggest improvements and potential enhancements.
Co2		Summary	Summarize the key takeaways from the evaluation.

The competition of steps marked with a * depends on the competition of previous steps: they may have to be left blank. Additionally, depending on the type of ontology considered (In4) some steps should be focused on more than others. This is indicated in the following table, where the cell contents indicate if the given step must/should/may be focused on depending on the ontology type:

Steps to focus on depending on the ontology type			
Step identifier	TLO	DO	AO
Title	MUST	MUST	MUST
Location	MUST	MUST	MUST
Purpose	MUST	MUST	MUST
Generality	MUST	MUST	MUST
High Level Requirements	MUST	MUST	MUST
Competency questions	MUST	MUST	MUST
Metric testing	MAY	SHOULD	SHOULD
Structural good-practices	MUST	MUST	MUST
Logical Properties	MUST	MUST	MUST
Ontological properties	MUST	SHALL	MAY
Concept coverage	MUST	MUST	MAY
Terminology	MUST	MUST	SHOULD
Users	MAY	MUST	MUST
Fairness	SHOULD	MUST	MUST
Ontology Lifecycle	MUST	MUST	MUST
Suggestions	MUST	MUST	MUST
Summary	MUST	MUST	MUST

Finally, please, fill the "Source" column with the one of the following values, depending on who is answering the question:
• "S" if self-assessment by a (co-)developer
• "E" if assessment by an external evaluator
• "D" if the assessment is already present in a document that can be referred to (and in that case cite the document)
• "O" If the previous three possibilities do not cover the current case (and in that case mention what is the answer source)

Here for each step in the template a corresponding clarification is supplied, which gives an indication on how to compile the step and/or on how to better develop or update the ontology in relation to the topic of the step.

- In1 – The full title of the ontology, expanding any eventual acronym.
- In2 – Where the ontology can be accessed at the moment. It is better if the ontology can be freely accessed at the address given by a resolvable URL (e.g. http://www.w3.org/2006/time, for the OWL Time Ontology), but if the ontology is not available in such a way, or not at all, indicate it here.
- In3 – What is the ontology purpose, that is why it was developed. For instance, an ontology may be developed with the high-level goal of bettering the management of knowledge and/or data of some company. In that case briefly explain why and how the ontology can achieve this. In other cases proceed analogously.
- In4 – Ontologies can range from very light-weight vocabularies to complex foundational ontologies. E.g., top-level ontology, mid-level ontology, domain ontology, application ontology.
- Fu1 – Has the ontology been developed to answer a certain set of requirements? An discussion of possible requirements can be found in [How to Write and Use the Ontology Requirements Specification Document, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Boris Villazón-Terrazas; a corresponding template is available at https://github.com/oeg-upm/LOT-resources]. A typical division between requirements is functional (what can the ontology do, usually interpreted as what queries or inferences does the ontology support) or non-functional (what are the `general' characteristics of the ontology, e.g. maintainability, multi-lingual support, etc.), but on a high-level this difference may not always be expressed. If the ontology is detailed in some research paper, a common costume is to list a series of requirements at the start of the paper. In addition, some ontology-development methodologies expressively asks for a list of requirements in input.
- Fu2 – Discuss here if the ontology satisfies the high level requirements specified in Fu1. This discussion is of high level, so no standard way to carry it out exists. One could, e.g., convert the high level requirements into narrower requirements, easier to check and formalize (e.g. SPARQL competency questions), check the satisfaction of those instead, and argue that therefore the ontology satisfy the high level requirements.
- Fu3,4 – Competency questions are questions that the ontology should be able to answer to. They are usually expressed first in natural language, then translated in a formal language. At first they were intended to be a set of questions that should be entailed, using logical inference, by the ontology [M. S. Fox and M. Gruninger, "Ontologies for enterprise integration," in CoopIS, 1994, pp. 82–89.], but then shifted to mean a set of questions that could be translated into SPARQL query language in order to check the

ontology expressiveness. However, if interpreted in such a way, competency questions are more apt to enquire about asserted instance data only [Camila Bezerra, Fred Freitas, and Filipe Santana. 2013. Evaluating Ontologies with Competency Questions. In Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 03 (WI-IAT '13). IEEE Computer Society, USA, 284–285. https://doi.org/10.1109/WI-IAT.2013.199]. Examples of competency questions can be find e.g. in [Potoniec J, Wiśniewski D, Ławrynowicz A, Keet CM. Dataset of ontology competency questions to SPARQL-OWL queries translations. Data Brief. 2020 Jan 7;29:105098. doi: 10.1016/j.dib.2019.105098. PMID: 31989008; PMCID: PMC6971340]. If some competency questions have been formulated for the ontology put them here.

- Fu5 – Discuss if reasoning is useful or not to answer the functional requirements given in competency questions. If the ontology is intended to be just a vocabulary for annotations, it is reasonable that it does not make use of axioms at all, but if the ontology wants to also model a given (or many) domain(s) using some axioms, are these axioms useful in answering the competency questions?
- Fu6,7 – Sometimes competency questions can be used to evaluate just the expressiveness of an ontology, in the sense that the ontology is thought to have a good expressiveness if its signature (=its classes and properties) allows to formulate many/all questions from a given relevant list in a natural (=easy to read) way. If that is the case, then one may not even need to test the competency questions against data. On the other hand, if there is the expectation that the ontology should be queried to retrieve data, testing of actual query-answering capability may be critical.
- S1,2 – If the ontology has been serialized as a graph, e.g. an RDF or OWL graph, one can calculate many metrics based on the structure of the graph (e.g. the breadth, width, and breadth-to-width ratio of a taxonomy, the average number of subclasses for a given class, etc.). There are numerous aspects of ontology quality in general, and many of these have been linked to sets of these numerical metrics. However, there meaning carried by these scores is matter of debate. In any case, if scores have been calculated for some of such metrics, always discuss their meaning case by case, in the context of the ontology, and prefer relative comparisons (e.g. compare the scores calculated from the target ontology to those calculated from similar ontologies).
- S3 – Use OOPS! And discuss the ensuing evaluation.
- On1 – The specific language the ontology is expressed in. If the ontology is written in OWL one can use e.g. profilechecker (https://github.com/stain/profilechecker) to check which dialect of OWL the ontology is written in.
- On2,3 – The appropriateness of a language for a given ontology is a complex matter. Some basic considerations that should at least taken into account are: - The fact that the language is well-known. - Expressivity vs computability tradeoff: the more things the

ontology can express the more difficult reasoning and other tasks become, and vice-versa. - Close vs open world: open-world-assumption-languages (such as OWL) allow for continuous schema integration, but make data validation difficult. Vice-versa for closed-world-assumption-languages (such as SHACL).

- On4 – Check if the ontology is consistent using an appropriate tool, such as Protegé with some reasoning engine. The ontology has to be consistent, moreover, the reasoning engine can reveal otherwise-unseen errors in the ontology: they may become evident if nonsensical inferences arise, then, by looking at the proof of the inference, one can correct them.
- On5 – If the ontology is a logical theory, can logical inference be used to some purpose? If not, why was the ontology developed as such?
- On6 – Discuss the axioms present in the ontology, if any. Unfortunately, this has to be done manually by some human expert, and there is no standard way.
- On7 – Alignment with an upper ontology can also be achieved indirectly, by aligning the ontology to an ontology itself aligned with an upper ontology. Alignment among ontologies, up to the level of upper ontologies, in one of the key parts of the EU project OntoCommons, since such an alignment propagates good ontological practices, maximize ontology reuse, and spares e.g. domain experts from having to model very abstract patterns, among other reasons. Alignment itself, in its most basic level, consists in stating meaningful equivalence and subsumption axioms between the classes and properties of the ontologies to be aligned. Some decision trees have been produced, that should help simplifying the work, for example, the a decision tree to align the classes of an ontology to the upper-level ontologies may be found in [C. Maria Keet, An Introduction to Ontology Engineering, p.129]. Choosing which upper level ontology align to, or arguing why a given upper level ontology was chosen, is another complex activity. The difficulty may be lessened through the use of tools such as ONSET (http://www.meteck.org/files/onset/). In any case, since alignments between some of the most common top level ontologies are among the outputs of the OntoCommons project, in the future the choice of a top level ontology will be less problematic.
- On8 – The OntoClean methodology was developed to ensure good quality of taxonomies from an ontological point of view. Unfortunately it must be applied manually by an expert. However, if proper alignment with an upper ontology was carried out, then this step should become redundant.
- On9 – Same as On7 but for narrower-scope ontologies.
- On10 – One should always ensure, by e.g. carrying out a proper literature review, that all the ontologies that could have been reused have been reused. If some ontology describing a relevant domain already exists, but was not reused, explain why (e.g. the scope is too narrow, that ontology would not satisfy the requirements, etc.).

- On11,12 – Ontology quality is many-faced, coverage of the relevant domain is one of the most important aspects. Discuss if and why the ontology covers the relevant domain. One could e.g. refer to the requirements list and claim that, by covering all the requirements related to expressiveness (e.g. competency questions), the ontology covers the relevant domain. Another typical approach consists of comparing the ontology to a list of domain-terms and checking how many of the terms are included in the ontology. The term list itself may be hand-crafted by domain experts or automatically extracted from a corpus of relevant literature.
- On13 – List any relevant standard that the ontology conforms to here.
- On14,15,16,17 – Good quality annotation are important and allow for automated extraction of good quality ontology documentation through tools such as WIDOCO (https://dgarijo.github.io/Widoco/). For annotation guide see R5:
- Ex 1,2,3 – As part of the evaluation, the ontology should be evaluated by domain experts, ontology experts, and users (if the latter are different from the formers). There is no standard way to do this, but usually a questionnaire is supplied to the experts and the users. As an output of the OntoCommons project, it is being considered if and how to supply a standard way to carry out these steps.
- Fa1,2 – The ontology should be evaluated with respect to FAIRness principles. These five questions are just a remainder of some main aspects of FAIRness, but using any dedicated tool (such as FOOPS! - https://foops.linkeddata.es/about.html, Or O'FAIREe - https://github.com/agroportal/fairness, or FAIR-Checker - https://fair-checker.france-bioinformatique.fr/) will evaluate all of the FAIRness indicators, then one can just report the results here.
- Fa3,4,5 – These are just some key points related to the Accessibility (the "A" in FAIR) of the ontology: a full FAIRness evaluation given e.g. in the previous points will cover also these steps. These three steps highlight the need for an ontology to be openly shared (with a corresponding license). If the ontology contains sensitive data, please consider the possibility to share at least the schema. Additionally, long term accessibility of the ontology is another important aspect. If the ontology's developer are not able or willing to commit to long-term hosting of permanent URIs, then they may consider to use e.g. the W3C's Permanent Identifiers for the Web project (https://w3id.org/).
- Li1,2 – The lifecycle of the ontology should be explicitly planned for. That is, some well-defined personnel should be tasked with long-term maintenance of the ontology.
- Co1 – After having gone through all previous steps, have some critical issued appeared? If so, how one could adjust them in the future?
- Co2 – Summarize the key points of the evaluation

R5: LOT4OCES provides a special guideline on publishing the ontology stressing the maintenance of FAIR principles while publishing. Although ontology artefacts may be registered in many FAIR data repositories, LOT4OCES highly recommends publishing the ontology in a dedicated ontology repository, e.g., IndustryPortal[7], for industrial ontology, and Matportal for material science ontology in addition to general purpose repositories. These ontology repositories, especially IndustryPortal provide many additional functionalities to maintain the versions of the source file, mappings among ontologies, and services to promote the exploitation of the ontologies in the repository, along with a detailed provision of FAIR metadata and FAIRness evaluator. A profile of minimum metadata for ontologies is a highly debated and ongoing study in the FAIR community, still, LOT4OCES recommends MOD (Industryportal currently supports version 1.4) metadata profile being the most comprehensive metadata profile, for the time being until a more conclusive profile is decided. LOT4OCES recommends that ontologies should be published with good-quality metadata to the extent that the integrated FAIR evaluator can score the artefact to be FAIR (at least 240 credits by O'Faire). The evaluation of FAIRness in turn completes the evaluation of the ontology. It also mentions a special role called Ontology curator, a specialist in registering the ontology in the repository and assuring FAIRness, to guide and support ontology developers in this task.

4.2.4 Notes on availability and maintenance

The OCES framework, and specifically the ontology stack, is stored in a Github repository: https://github.com/OntoCommons/OntologyFramework. In the following table the folders and their contents are described.

Folder path	Content
owl	Contains owl ontology artefacts belonging to the OCES ontology stack
owl/TRO	Contains TLOs, mappings among TLOs, common annotation file, and sample TRO module (tro.owl)
owl/TRO/TLOs	Contains TLO owl files (e.g., BFO, DOLCE, EMMO) and sample tlo module (tlo.owl)
owl/TRO/META	Contains TLO mapping files and sample META module (meta.owl)
owl/MRO	Contains bridge concets, MLO mappings and sample mro module (mro.owl)

[7] http://industryportal.enit.fr/

owl/CONCEPTS	Contains bridge concepts in owl format (includes mapping to TLOs)
owl/doc	Contains bridge concept analysis in template (markdown files)
owl/MLO	Contains MLO owl files
owl/MLO-Mappings	Contains mappings of MLO concepts to bridge concepts
owl/DLO	Contains DLO ontologies and their mappings to MLOs
owl/doc	Contains various methodological guidelines, technical principles, and FOL mapping

LOT4OCES also provides a one-stop-shop alike system for orchestrating ontology development activities available at https://tooling.ontocommons.linkeddata.es/ depicted in Figure 18. this web application allows uploading one or more ontologies and execute a number of tools over the ontologies. More precisely, the tools available are Widoco (https://dgarijo.github.io/Widoco/) for ontology documentation, OOPS (https://oops.linkeddata.es/) and Themis (http://themis.linkeddata.es/) for ontology evaluation and Astrea (https://astrea.linkeddata.es/) for Shapes generation. the pipeline could be executed over OWL files or could be triggered by ontology conceptualisations following the Chowlk notation, in which case the Chowlk converter will be executed in order to obtain an OWL ontology to be used as input for the rest of tools. This system also provide links to main ontology registries to look for ontologies or to registered the ontology being built as well to links to best practices and recommendations for publishing FAIR vocabularies.

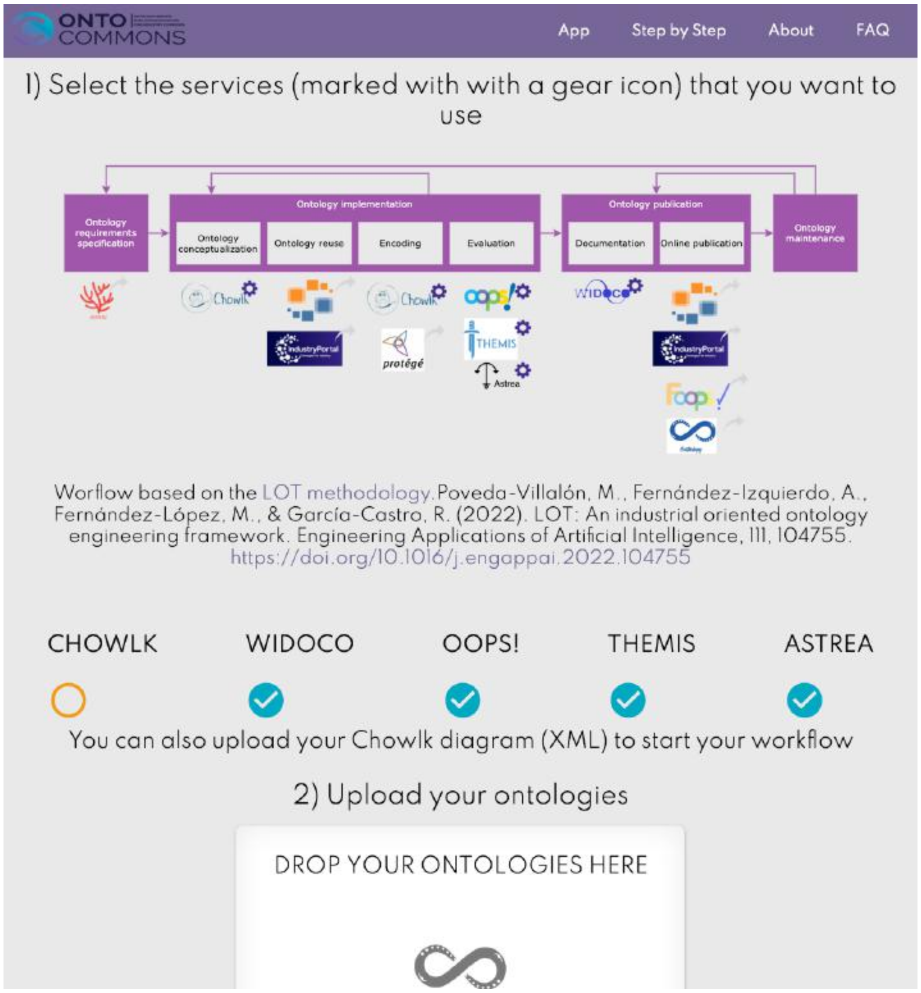

Figure 18 - LOT4OCES ontology development tooling

5. Technical Principles and Best Practices

In the following subsections a list of technical Principles and best Practices endorsed by the OntoCommons Consortium is reported. Users can find both requirements and recommendations they should follow in order to adhere to the OCES, and improve their knowledge representation artifacts, doubling as a list of explicit normative commitments. The Technical Principles take inspiration from the relevant literature and de facto standards, contributing towards standardization by favouring the pervasiveness of existing solutions.

OCES identifies the following topics under the technical principle. However, during the course of D2.6, three major topics: IRI conventions, metadata convention, and FAIRness of ontology is developed. In this section we present only these three topics. The future effort should make effort in developing the conventions on the rest of the topics.

Topics of technical principles

- **IRI and naming conventions**

What will be the domain and identifier for the IRI and URI? The naming convention for the labels for the terms (both class and relation) needs to be standardized. for example,OverlappedBy vs Overlapped by vs overlappedBy vs overlapped-by

- **Metadata for ontology content and FAIRness of ontology**

(annotation properties) What are the minimum and full list of available metadata to be associated with the ontology and its contents (terms, axioms)? Some popular collections are given by owl, rdf, dcat, skos, dcterm, mod.

- **Language and expressivity**

A supported list of language need to be specified for alignment and development. Though most of the ontologies in MLO and DLO are in OWL, the level of expressivity (OWL Lite/DL/Full) for different purposes of ontology need to be classified. For harmonizing TLOs, the level of rigorousness in logical alignment needs to be specified. For such, a suitable language from Common Logic, CASL may need to be specified.

- **Reasoner and prover**

Specify which reasoner(s) will be used. For example, OWL-DL specific: Pellet, Hermit, Fact++ etc. FOL specific: Prover9 and Hets family.

- **Serialization format**

What will be the serialization format in which ontology files will be encoded? For example, OWL/XML, RDF/XML, JSON-LD, Turtle).

- **Mapping relation vocabulary**

Which set of relations will be used for mapping. owl:EquivalentClass, SubClass are stricter. Also, SKOS provides a set of mapping relationship e.g., skos:closeMatch,skos:exactMatch, skos:broadMatch, skos:narrowMatch and skos:relatedMatch.

- **Versioning scheme**

How the ontology files will be versioned? What will be the numbering pattern?

- **Documentation and visualization**

How an ontology (or alignment) will require to be documented (structure)? What will be the format, e.g., doc, latex, markdown? Relevant tools, e.g., LODE, widoco? Ontology documentation should include a visual for the model in a specified format, e.g. UML, CHOWLK, or WebVOWL.

- **Tools, management, review process**

For collaborative development, own cloud is not suitable for development. Common development phases need appropriate tools. Following are just suggestions for typical software development.

- Collaboration (Confluence)
- Source code repository (Github)
- CI/CD (Jenkins)
- Issue tracking (JIRA)

How to review, validate and approve development from different members. We may need to set up a review board that will approve development commits for merging. A set of clear quality criteria may also need to be defined.

5.1 IRI convention

R1: Every Ontology Artefact developed under OCES and every Term it contains must be identified by an Ontology Common IRI, or OCIRI.

The FAIR principles, under the Findability and the Accessibility chapters respectively, state that: F1. (Meta)data are assigned a globally unique and persistent identifier. A1. (Meta)data are retrievable by their identifier using a standardised communications protocol.

R1.1: An OCIRI must be well-formed, that it must follow RFC3987 [1] from The Internet Engineering Task Force (IETF).

the Internationalized Resource Identifier (IRI), as a complement to the Uniform Resource Identifier (URI). An IRI is a sequence of characters from the Universal Character Set (Unicode/ISO 10646). A mapping from IRIs to URIs is defined, which means that IRIs can be used instead of URIs, where appropriate, to identify resources.

R1.1.1: An OCIRI must have the following grammar.

The grammar adheres to the IRI grammar in RFC3987 but specializes for referring to Ontology artefacts. The grammar introduces some non-terminals prefixed with 'o' to distinguish them from the original non-terminals used in the IRI grammar in RFC3987. All terms which are capitalized and prefixed by 'OC' are especially recommended for ontology artefacts developed part of OCES and standardized further. All other non-terminals refer to the non-terminals used in the IRI grammar in RFC3987.

OCIRI ::= scheme ':' oauthority opath ofragment

oauthority ::= '//' (OCHOST | ihost)

opath ::= (opath-absolute)? opath-relative

opath-absolute ::= ('/' isegment)*

opath-relative ::= ('/' isegment)*

ofragment ::= '#' ifragment

OCHOST ::= ohost (':' port)? ('/' opath-absolute)?

ohost ::= (IP-literal | IPv4address | oreg-name)

oreg-name ::= (iunreserved | pct-encoded | sub-delims)+

A domain ontology called 'plastonto' of version 3 beta http://purl.ontocommons.eu/ontology/dlo/srao-0000211/plastonto/3/beta

A class 'Plastic' in a domain ontology called 'plastonto' http://purl.ontocommons.eu/ontology/dlo/srao-0000211/plastonto#oxcy4f

A mapping file between DOLCE (source) to BFO (target) of version 0.2 alpha http://purl.ontocommons.eu/mapping/dolce/bfo/cnrfol/0.2/alpha

Top reference ontology file http://purl.ontocommons.eu/TRO/bfo-dolce-emmo

R1.1.1: OntoCommons project MUST choose a single authority for all ontology artefacts released by the OntoCommons.

Note: It is not necessary for an ontology to follow the strategy of OntoCommons for obtaining authority to be part of OCES.

OCHOST
ontocommons.eu Currently no resolver or handler is installed.
purl.ontocommons.eu PURLs are Web addresses or Uniform Resource Locators (URLs) that act as permanent identifiers in the face of a dynamic and changing Web infrastructure. Instead of resolving directly to Web resources (documents, data, services, people, etc.) PURLs provide a level of

indirection that allows the underlying Web addresses of resources to change over time without negatively affecting systems that depend on them. This capability provides continuity of references to network resources that may migrate from machine to machine for business, social or technical reasons. Example handler: https://github.com/OBOFoundry/purl.obolibrary.org provides tools for managing OBO Foundry Permanent URLs (PURLs). Like w3id they use per-directory Apache configuration files (.htaccess files), each of which uses RedirectMatch directives to redirect PURL requests to their proper targets.
w3id.org/ontocommons The purpose of this https://w3id.org/ is to provide a secure, permanent URL re-direction service for Web applications. This service is run by the W3C Permanent Identifier Community Group. Web applications that deal with Linked Data often need to specify and use URLs that are very stable. They utilize services such as this one to ensure that applications using their URLs will always be re-directed to a working website. This website operates like a switchboard, connecting requests for information with the true location of the information on the Web. The switchboard can be reconfigured to point to a new location if the old location stops working. For more information, see https://github.com/perma-id/w3id.org.
doi.org/10.1000/182 International DOI Foundation designed the DOI system to provide a form of persistent identification, in which each DOI name permanently and unambiguously identifies the object to which it is associated (although when the publisher of a journal changes, sometimes all the DOIs will be changed, with the old DOIs no longer working). It also associates metadata with objects, allowing it to provide users with relevant pieces of information about the objects and their relationships. Included as part of this metadata are network actions that allow DOI names to be resolved to web locations where the objects they describe can be found. To achieve its goals, the DOI system combines the Handle System and the indecs Content Model with a social infrastructure. DOI registration is not free and need annual fee for maintenance. See https://www.medra.org/ (One of the DOI service provider) for more information.
https://n2t.net/ark:/12345 or https://ontocommons.eu/ark:/12345 An **Archival Resource Key** (**ARK**) is a multi-purpose URL suited to being a persistent identifier for information objects of any type. It is widely used by libraries, data centers, archives, museums, publishers, and government agencies to provide reliable references to scholarly, scientific, and cultural objects. In 2019 it was registered as a Uniform Resource Identifier (URI). ARKs can be maintained and resolved locally using open source software such as Noid (Nice Opaque Identifiers) or via services such as EZID. Most implementations are decentralized and no fees are charged for the right to assign ARKs. Some implementations choose to publish ARKs via the centralized N2T (Name-to-Thing) resolver.

Figure 19 - OCHOST specifications

R1.1.1a: An OCHOST is the authority that SHOULD be domain administered and owned by the OntoCommons project or the project adopting the OCES.

R1.1.2: OCES network of ontologies should follow the generality of the scope of the ontology and OCES dependency structure to determine the opath-absolute.

http://purl.ontocommons.eu/tro/ is a dynamically created import file importing

one of

http://purl.ontocommons.eu/meta/dolce2bfo

that imports dolce and bfo

http://purl.ontocommons.eu/tro/ is a dynamically created import file importing

one of

http://purl.ontocommons.eu/meta/emmo2bfo

that imports emmo and bfo

R1.1.2a (exception): If an **existing ontology** is included in the network then the default IRI of the ontology MAY be used without modification.

R1.1.2b: The first-place token for opath-absolute **should** denote the type of the artifact, i.e., one of tro,mro, as they are defined below.

ontology: an artifact with *at least one* owl class or property.

mapping: an artifact with *only* mapping assertions between classes, which are not part of the artifact.

import: an artifact with *only* import statements and no class or property assertion or mapping assertion.

R1.1.2c: If an artifact is an ontology, then the token at the second place for opath-absolute **should** denote the generality of the ontology i.e., one of tlo, mlo, dlo, alo, data, as per the following definitions:

Refer to the formal definitions of TLO, MLO, DLO definitions in D2.5 beta. However, following rules of thumb may be used for quick classification:

- All existing TLO and MLOs are covered in state-of-the-art published by WP2 of OntoCommons.
- A new TLO may need community consensus to be regarded as a TLO.
- A new MLO should cover one of the topics for MLO listed in D2.5.

- An existing or new DLO should have at least one of the topic area in DFG, IERC or SRAO classification as its scope.
- An existing or new ALO should have some application (a business process or a software) as its scope created to annotate data only relevant to that application.

R1.1.2c: If an artifact is an ontology, then the token at the second place for opath-absolute **should** denote the taxonomic position of the topic the ontology covers, as per the OCES accepted subject taxonomy. For encoding the token, the CURIE of the source of the subject taxonomy **may** be followed by the subject ID as per the classification scheme of the subject taxonomy chosen with the separator '_'.

> **The** opath-absolute **for a domain ontology for 'plastic engineering' is ...dlo/ontology/DFG_401-04/... (following DFG classification) or ...dlo/SRAO_0000211/... (following SRAO classification)**

R1.1.2d: if an artifact is a mapping, then the token at the second place must mention the source ontology and the third place must mention the target ontology. Only the acronym of the ontology should be used.

> **The** opath-absolute **for a mapping artifact containing mappings from DOLCE to BFO is ...mapping/DOLCE/BFO/...**

R1.1.2e: If an artifact is import, then the token at the second place for opath-absolute is determined depending on whether the IRI is transparent or opaque.

5.1.1 ifragment (transparent term IRI)

Class Names

Class names **MUST** be given in Upper Camel Case, each word capitalized, and no separation or punctuation between words. As with the module names, no acronyms **MUST NOT** occur except those in the dictionary, such as RADAR.

- .../SupplyChainReferenceOntology/SupplyChainShippingProcess

The versioned forms are as follows:

- .../SupplyChainReferenceOntology/SupplyChainShippingProcess

Property Names (Relations)

All property names **MUST** be in lower Camel Case, the first word lower case and each subsequent word capitalized with no separation or punctuation between words.

Object Properties

All object property names **MUST** be verbs or a verb phrase. For example:

- hasParticipant
- participatesIn

Data Properties

A data property **MUST** be a verb phrase starting with is for boolean (true/false) or has for any other data type. The data property **SHOULD** end with Value.

Examples:

- hasTagValue
- hasDateValue
- isTransferable

5.1.2 ifragment (opaque term IRI)

ifragment (opaque term IRI)
Globally Unique Identifier (GUID) GUID was a term first used by Microsoft to refer to a specific variant of a similar term, Universally Unique Identifier, or UUID. Since then, the terms have been combined, with the RFC 4122 specification using them synonymously. Different versions of GUIDs follow the RFC 4122 specification. GUIDs are constructed in a sequence of digits that equal 128 bits. The ID is in hexadecimal digits, meaning it uses the numbers 0 through 9 and letters A through F. The hexadecimal digits are grouped in a format that is 36 characters long -- 32 hexadecimal characters grouped as 8-4-4-4-12 and separated by four hyphens: {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}.
UUID Similar to GUID
ShortID https://github.com/bolorundurowb/shortid
OBO ID Each OBO ID is assigned to a only single term within the set of all OBO ontologies. There is a 1:1 mapping of OBO IDs to Foundry-compliant URIs. Identifier Syntax. Identifiers (IDs) in OBO should be strings consisting of an IDSpace concatenated to a LocalID via a : (colon) character.

Figure 20 - ifragment Table (opaque IRI)

5.2 Annotation and metadata guide

The set of metadata recommended in this document follows several metadata standards, e.g., FAIR principles, IOF Annotation guideline as well as considers stakeholder's input collected by OntoCommons. Being true to the pluralistic approach of OCES, the following recommendation harmonizes several existing metadata standards. The metadata standard and guideline recommended in this document should be available as annotation properties for sake of being serialized as part of the source file. The metadata vocabulary to be used by an ontology registry or repository to organize ontology artifacts are not covered, however, may be derived from this convention. The recommended metadata does not have any domain-specific semantics attached to them and may be applied to ontology from any domain alike.

5.2.1 Rules for development, selection, and maintenance of annotation properties

OCES provides an extensive set of annotation properties to be used as metadata to annotate both an ontology and its content (class, properties, datatype, axioms) harmonizing existing metadata vocabularies (e.g., RDF, SKOS, OMV, DCAT, MOD etc.). This section provides the guideline to develop and maintain such set of annotation properties.

OCES does not restrict ontology developers to add new annotation properties to the recommended set. This may be required for specializing an annotation property for their need. For example, a new user role 'reviewer' may need to be mentioned. In that case, the rules in this section SHOULD be followed for the development and maintenance of such customized set of annotation properties.

1. Every metadata MUST be available as an OWL annotation property.

Annotation properties are declared using OWL 2.0 syntax to take full advantage of enhanced annotation capabilities[8][1]. E.g., annotation of annotations themselves and some annotation properties are restricted to be used for annotating ontology header while others for terms. Please note that these restrictions have no semantic meaning in the OWL 2 Direct Semantics but carry the standard RDF semantics in the RDF-based Semantics (via the mapping to RDF vocabulary).

2. All annotation properties MUST be curated in a single and stand-alone (contains only annotation properties) ontology artifact for ease of reuse.

All annotation properties are curated in http://www.purl.ontocommons.edu/tro/common-annotation with CURIE common-ap.

3. Each annotation property MUST have the annotations as listed in the following table.

[8] https://www.w3.org/TR/owl2-overview/#Relationship_to_OWL_1.

Annotation Property	Usage
rdfs:label	Name of the given annotation property in a human-readable form. Example:
skos:definition	Statement or formal explanation of the meaning of the given annotation property. Example:
lof-av:usageNote	Natural Language explanation about how the given annotation property is to be used.

Figure 21 - Required Annotations

4. Each annotation property MAY have the annotations as listed in the following table.

Annotation Property	Usage
rdfs:isDefinedBy	Original resource defining the given annotation property (required when curated, see R 1.2.1). Example:
skos:closeMatch	Other annotation properties which can be used interchangeably with the given annotation property.
skos:example	Example of the use of the given annotation property

Figure 22 - Optional Annotations

5. The original source (IRI) of an annotation property SHOULD be given with annotation rdfs:isDefinedBy if the corresponding ontology artifact is not directly imported by common-ap.
6. If an annotation property is curated from an existing metadata vocabulary that does not explicitly define the metadata as annotation property, then the annotation property SHOULD be redefined[9][2] in common-ap with the original IRI annotated using rdfs:isDefinedBy.
7. If an annotation property is curated from an existing set of annotation properties, then the original IRI SHOULD be reused as is.
8. An annotation property MAY have one or more domain from owl:Ontology, owl:Class, owl:ObjectProperty, owl: DatatypeProperty, owl: NamedIndividual, owl:Axiom to restrict the use of annotation property to certain ontology elements.
9. An annotation property SHOULD have a datatype as its range to restrict the type of value given for the annotation property.

[9] It is required to avoid conflicts with original ontology as the sets of object properties, datatype properties, annotation properties and ontology properties must be mutually disjoint. OWL Full allows object properties and data properties may not be disjoint.

10. Only xsd datatypes SHOULD be used as range restriction for an annotation property.

5.2.2 Rules for annotating a terms

1. Each term MUST have at least one annotation for the following requirements as listed in the following table.

Req.ID	Requirement	Recommended annotation properties
Req 1	A term needs to have at least one human-readable name of the concept that it represents.	rdfs:label, skos:prefLabel
Req 2	A term needs to have a textual definition.	skos:definition
Req 3	A term needs to provide the source IRI of a class from another ontology, which it is adapted/inspired from.	skos:closeMatch, iof-av:adaptedFrom, iof-av:directSource, rdfs:isDefinedBy dc:source, dcterms:source

Figure 23 - Requirements for Class Annotations 1

The annotation value for Req 1 SHOULD be in natural language with corresponding language tag.

2. The annotation value for Req 1 MUST NOT be an abbreviation.
3. An acronym or initialism SHOULD be avoided as the annotation value for Req 1. See Req 1.2.
4. The annotation value for Req 1 MAY be an acronym or initialism if such is more popular than the full form name. e.g., DVD, RADAR.
5. More than one annotation MUST NOT be used for Req 1.
6. The annotation value for Req 1 SHOULD be unique in the scope of the parent ontology.
7. The annotation value for Req 2 SHOULD be derived from a trusted source, e.g., standard, book, manual.
8. The source of the annotation value for Req 2 MUST be given following the *citation rules*.
9. If the annotation value for Req 2 is not derived from an external source but constructed by the ontology developers, then SHOULD follow *ISO 704 Terminology work — Principles and methods*[10[3]].
10. The annotation value for Req 2 MUST NOT use an article (The/A/An) before the concept (label) being defined.
11. The annotation value for Req 2 MUST NOT be circular.
12. The annotation value for Req 2 MUST NOT include the annotation value for Req 1 for the same class.

10[3] https://www.iso.org/standard/79077.html

skos:definition for Mechanical Arm:
Incorrect: Mechanical arm is a robotic ….
Correct: Robotic instrument that

21. If another term (or its label, i.e., the annotation value for Req 1) from the parent or an external ontology is used then the term MUST be distinguished from the rest of the text with a parentheses and appropriate prefix.
22. The annotation value for Req 2 MUST NOT include an IRI, or CURIE. Instead use the label of the ontological term.
23. The annotation value for Req 2 SHOULD NOT contain an acronym or initialism if such is more popular than the full form name. e.g., DVD, RADAR.
24. The class SHOULD NOT provide the source IRI of a class from another ontology, which it is copied or adopted from if the latter ontology is imported or original IRI is preserved.

shipment preparation process: planned process in which some "bfo: material entities" are prepared to be transported together to a receiver's location

postal address: designation of a location (site) to which mail is delivered

21. Each term SHOULD have at least one annotation for the following requirements as listed in the following table.
22. The annotation value of Req 2.1 SHOULD be a Well-Formed formula.
23. A standard notation MUST be used for writing the formula. Example, CLIF, TPTP, Prover 9.
24. The notation used for writing the formula MAY be added as annotation of the annotation for Req 2.1. Please see language specification rule.
25. More than one annotation for Req 2.1 SHOULD NOT be used.

Req.ID	Requirement	Recommended annotation properties
Req 2.1	A term needs to have a logical definition.	iof-av:firstOrderLogicDefinition
Req 4	A term needs to provide example(s) of its extensions.	skos:example, vann:example
Req 5	A term needs to provide relevant development status	iof-av:maturity, owl:deprecated, sw:term_status, obo:IAO_0000114

Figure 24 - Requirements for Class Annotations 2

31. Each class MAY provide annotation for the following requirements as listed in the following table.

Req.ID	Requirement	Recommended annotation properties
Req 1.1	A class needs to provide alternative names for the concept it represents.	skos:altLabel, iof-av:synonym
Req 1.2	A class needs to provide alternative names in different formats, e.g., acronym, abbreviation, initialism.	iof-av:symbol | iof-av:abbreviation | iof-av:acronym
Req 2.2	A class needs to provide logical statements for further constraining the interpretation of the class or as a theorem involving the class.	iof-av:logicAxiom
Req 6	A class needs to provide additional references for the concept it represents.	rdfs:seeAlso
Req 6.1	A class needs to provide additional information in the form of reference, explanatory text, usage example, or scope restriction.	iof-av:explanatoryNote | iof-av:usageNote | skos:scopeNote | vaem:rationale | emmo:elucidation | emmo:conceptualisation
Req 7	A class needs to mention the person or organization who performed some role in the creation and edition.	dcterms:creator, dc:creator, schema:creator, pav:createdBy prov:wasAttributedTo, dcterms:contributor, dc:contributor, schema:contributor pav:contributedBy

Figure 25 - Requirements for Class Annotations 3

5.3 FAIR metadata

5.3.1 Key FAIR properties for naming a semantic artifact

Mandatory (M) category:

Metadata property used for **"acronym"** (1)	
Voted FAIR property	mod:acronym
All other possible recognized FAIR properties	mod: acronym

5.3.2 Key FAIR properties for identifying a semantic artifact

Metadata property used for " URI" (1)	
Voted FAIR property	omv:uri
All other possible recognized FAIR properties	none
FAIR principle(s)	FINDABLE

Metadata property used for "other identifier"	
Voted FAIR property	dcterms:identifier
All other possible recognized FAIR properties	dc:identifier, skos:notation, adms:identifier, schema:identifier
FAIR principle(s)	FINDABLE

5.3.3 Key FAIR properties for versioning a semantic artifact

Mandatory category:

Metadata property used for "Version IRI"	
Voted FAIR property	owl:versionIRI
All other possible recognized FAIR properties	none
FAIR principle(s)	FINDABLE

Recommended category

Metadata property used for "Previous version"	
Voted FAIR property	omv:hasPriorVersion
All other possible recognized FAIR properties	owl:priorVersion, dct:isVersionOf, door:priorVersion, prov:wasRevisionOf, adms:prev, pav:previousVersion, pav:hasEarlierVersion, owl:PriorVersion

FAIR principle(s)	FINDABLE

5.3.4 Key FAIR properties for documenting a semantic artifact

Mandatory category:

Metadata property used for " format "	
Voted FAIR property	dcterms:isFormatOf
All other possible recognized FAIR properties	mod1.1:syntax
FAIR principle(s)	REUSABLE

Metadata property used for " accessURL "	
Voted FAIR property	dcat:accessURL
All other possible recognized FAIR properties	doap: download-page , omv:resourceLocator
FAIR principle(s)	REUSABLE

Metadata property used for " license "	
Voted FAIR property	dct: license
All other possible recognized FAIR properties	dc: rights, dct: rights, dct: license, cc: license, schema:license
FAIR principle(s)	REUSABLE

Metadata property used for " type "	
Voted FAIR property	dcterms:type
All other possible recognized FAIR properties	omv:isOfType

FAIR principle(s)	REUSABLE

Metadata property used for " landingPage "	
Voted FAIR property	foaf:homepage
All other possible recognized FAIR properties	cc:attributionURL, schema:mainEntityOfPage, doap:log, mod1.0:homepage
FAIR principle(s)	REUSABLE

Metadata property used for " creator"	
Voted FAIR property	dct:creator
All other possible recognized FAIR properties	dc:creator, dct:creator, foaf:maker, prov:wasAttributedTo, doap:maintainer, pav:authoredBy, pav:createdBy, schema:author, schema:creator
FAIR principle(s)	REUSABLE

Metadata property used for "Creation date"	
Voted FAIR property	dcterms:created
All other possible recognized FAIR properties	omv:creationDate, dct:date, dct:issued, mod:creationDate, doap:created, schema:dateCreated, prov:generatedAtTime, pav:createdOn, pav:authoredOn, pav:contributedOn, oboInOwl:date, oboInOwl:hasDate
FAIR principle(s)	REUSABLE

Metadata property used for "Description"	
Voted FAIR property	omv:description
All other possible recognized FAIR properties	dct:description, rdfs:comment, doap:description, schema:description, oboInOwl:remark

FAIR principle(s)	REUSABLE

Metadata property used for "Keyword"	
Voted FAIR property	omv:keywords
All other possible recognized FAIR properties	schema:k eywords, http://www.isibang.ac.in/ns/mod/1.1/keyword, dcat:keyword, dcterms:keywords
FAIR principle(s)	REUSABLE

Metadata property used for " subject "	
Voted FAIR property	omv:hasDomain
All other possible recognized FAIR properties	dcterms:subject, schema:about, foaf:topic
FAIR principle(s)	REUSABLE

Recommended category

Metadata property used for " usedOntologyEngineeringMethodology"	
Voted FAIR property	omv:usedOntologyEngineeringMethodology
All other possible recognized FAIR properties	schema:publishingPrinciples,mod1.1:methodologyUsed, adms:representationTechnique
FAIR principle(s)	REUSABLE

Metadata property used for "definition Property"	
Voted FAIR property	mod:definitionProperty

All other possible recognized FAIR properties	none
FAIR principle(s)	REUSABLE

Metadata property used for “downloadURL”	
Voted FAIR property	dcat:downloadURL
All other possible recognized FAIR properties	schema:distribution, doap:download-mirror , void:dataDump
FAIR principle(s)	REUSABLE

Metadata property used for “hasFormalityLevel”	
Voted FAIR property	omv:hasFormalityLevel
All other possible recognized FAIR properties	mod1.0:formalityLevel
FAIR principle(s)	REUSABLE

Metadata property used for “dcterms:accrualMethod”	
Voted FAIR property	dcterms:accrualMethod
All other possible recognized FAIR properties	none
FAIR principle(s)	REUSABLE

Metadata property used for “ Contributor”	
Voted FAIR property	omv:hasContributor
All other possible recognized FAIR properties	dc:contributor, dct:contributor, doap:helper, schema:contributor, pav:contributedBy, oboInOwl:savedBy

FAIR principle(s)	REUSABLE

Metadata property used for "Citation"	
Voted FAIR property	dcterms:bibliographiccitation
All other possible recognized FAIR properties	omv:reference, dct:bibliographicCitation, foaf:isPrimaryTopicOf, schema:citation, cito:citesAsAuthority, schema:citation
FAIR principle(s)	REUSABLE

Metadata property used for " Prefix"	
Voted FAIR property	vann:preferredNamespacePrefix
All other possible recognized FAIR properties	idot:preferredPrefix, oboInOwl:default-namespace, oboInOwl:hasDefaultNamespace
FAIR principle(s)	REUSABLE

Metadata property used for " Namespace URI"	
Voted FAIR property	vann:preferredNamespaceUri
All other possible recognized FAIR properties	void:uriSpace

Metadata property used for " synonymProperty"	
Voted FAIR property	mod:synonymProperty
All other possible recognized FAIR properties	none

Metadata property used for "prefLabelProperty"

Voted FAIR property	mod:prefLabelProperty
All other possible recognized FAIR properties	none

Metadata property used for " conforms to"	
Voted FAIR property	omv:conformsToKnowledgeRepresentationParadigm
All other possible recognized FAIR properties	dcterms:conformsTo

Metadata property used for " language"	
Voted FAIR property	omv:hasOntologyLanguage
All other possible recognized FAIR properties	schema:fileFormat, mod1.1:ontologyDesignLanguage

Metadata property used for "rights"	
Voted FAIR property	dcterms:accessRights
All other possible recognized FAIR properties	mod11:accessibility, dcterms:accessRights

Optional category

Metadata property used for " Abstract"	
Voted FAIR property	dcterms:abstract
All other possible recognized FAIR properties	none

Metadata property used for " Status"	
Voted FAIR property	omv:status
All other possible recognized FAIR properties	adms:status, idot:state

Metadata property used for " usedEngineeringTool"	
Voted FAIR property	omv:usedOntologyEngineeringTool
All other possible recognized FAIR properties	pav:createdWith, oboInOwl:auto-generated-by, mod:1.1toolUsed,

Metadata property used for "policy"	
Voted FAIR property	dcterms:relation
All other possible recognized FAIR properties	door:ontologyRelatedTo

Metadata property used for "relation"	
Voted FAIR property	dcterms:accrualPolicy
All other possible recognized FAIR properties	none

Metadata property used for "includedInDataCatalog"	
Voted FAIR property	schema:includedInDataCatalog
All other possible recognized FAIR properties	none

Metadata property used for "competencyQuestion"

Voted FAIR property	mod:competencyQuestion
All other possible recognized FAIR properties	none

Metadata property used for "accrualPeriodicity"	
Voted FAIR property	dcterms:accrualPeriodicity
All other possible recognized FAIR properties	none

Metadata property used for "spatial"	
Voted FAIR property	dcterms:coverage
All other possible recognized FAIR properties	schema:spatial

Metadata property used for "was generated by"	
Voted FAIR property	prov:wasGeneratedBy
All other possible recognized FAIR properties	none

Metadata property used for " Backward Compatible"	
Voted FAIR property	omv:isBackwardCompatibleWith
All other possible recognized FAIR properties	door:backwardCompatibleWith, owl:BackwardCompatibleWith

Metadata property used for " Incompatibilty"	
Voted FAIR property	omv:isIncompatibleWith

All other possible recognized FAIR properties	owl:incompatibleWith, door:owlIncompatibleWith

Metadata property used for "Modification Date"	
Voted FAIR property	dcterms:modified
All other possible recognized FAIR properties	dct:modified, schema:dateModified, pav:lastUpdateOn, mod:updated

Metadata property used for "Source"	
Voted FAIR property	dcterms:source
All other possible recognized FAIR properties	dcterms:source, schema:isBasedOn, prov:isDerivedFrom, prov:wadInfluencedBy nkos:basedOn pav:derivedFrom

Metadata property used for "Publisher"	
Voted FAIR property	dcterms:publisher
All other possible recognized FAIR properties	schema: publisher, adms:schemaAgency

Metadata property used for "logo"	
Voted FAIR property	foaf:logo
All other possible recognized FAIR properties	schema:logo

Metadata property used for "Depiction"	
Voted FAIR property	foaf:depiction

All other possible recognized FAIR properties	schema:image, doap:screenshots

6. Concluding Remarks

D2.9 marks the culmination of WP2's collaborative efforts throughout the OntoCommons Project, representing a pivotal achievement resulting from extensive partnerships with other WPs and external experts. This deliverable stands as the primary objective of T2.6 and holds a foundational role in realizing the goals set forth in O2.1 and O2.4. Aligned with the DoA's specifications, D2.9 takes on the format of a white paper, serving as both a user manual and a guide for maintenance and development. It provides a wealth of best practices and guidelines, aimed at facilitating the utilization, exploitation, maintenance, and further evolution of the tools, methodologies, and infrastructure derived from WP2's endeavours.

While comprehending the guiding principles, theoretical assumptions, and practical considerations that have guided WP2's work is vital for fully achieving the objectives outlined above, it's important to note that D2.9 predominantly emphasizes the practical aspects. It furnishes clear and actionable workflows designed to attain specific outcomes, ensuring that stakeholders with can find valuable guidance tailored to their precise needs no matter how, and to what degree, they decide to interact with the OCES.

To establish a unique point of reference for stakeholders seeking documentation related to the OCES formal framework, D2.9 furnishes a comprehensive summary of the results generated in the context of T2.4 and T2.5. In particular, it offers in-depth elaboration on key aspects, marking the first comprehensive aggregation of material concerning Bridge Concepts.

Nevertheless, D2.9's main concern remains expounding the OCES formal framework to users and future developers. D2.9’s substantial contributions encompass a thorough exploration of the framework’s core machinery, with a specific focus on the OWL implementation, coupled with a comprehensive list of technical principles endorsed by OntoCommons. These principles serve as both practical recommendations, grounded in the current landscape and state-of-the-art literature, and as essential requirements to guide stakeholders, further promoting interoperability via standardization where possible.

This white-paper should provide all the information necessary in the first phase of deployment of the OCES. That said, field-tests were impossible in the course of the project. A proper manual should thus be created taking into account empirical data and feedback at a later stage, taking this as a starting point. It might then be useful to divide the material depending on the various sections’ target readers, possibly resulting in separated users and maintenance manuals. Foundational and technical points should arguably never be omitted, though the more complex aspects (beyond what has been discussed in this document) should find their place in specialistic papers.

The OCES framework is currently functional, and it is expected that this document will have a positive impact in advancing its successful exploitation, contributing to the overarching goals of the OntoCommons Project, in the context of all the relevant EU initiatives.