Ontology Ecosystem Specification

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

Ontology Ecosystem Specification

Lead author	Mathieu d'Aquin (NUIG)
Contributors	Alba Fernández-Izquierdo (UPM), María Poveda-Villalón (UPM), Andrew Dubber (ICF), Michela Magas (ICF), Yann Le Franc (eSDF)
Peer reviewers	Hedi Karray (ENIT)
Version	Final
Date	07/06/2021

Keywords

Ontology; Ontology Engineering Tools; Ontology Services;

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 in order 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 as a result of using its content.

Executive Summary

This document reports on the first focused workshop aimed at the definition, extent, characteristics and components of an ontology ecosystem toolkit. We first provide an overview of the ontology engineering and use process, as context for the analysis of the material collected during the workshop and of other sources of requirements. The workshop itself took place on the 19th of March 2021 and involved over 100 participants. It included 11 presentations and a panel. As a way to collect information from the ontology engineering community, a questionnaire, four polls, notes and moderation of the written discussions during the workshop were used. This allowed us to better understand what tools exists to support the process creating and using ontologies in 2021, what tasks and methods were particularly important, and what gaps existed in current toolkits. As a result of this, and taking into account the broader context of the project, we provide an overview of the components to be included in the ontology ecosystem toolkit as a set of interacting services supporting the five phases of the project (requirement specification, implementation, publication, maintenance, use). We also present additional, mostly non-functional requirements related to the collaborative aspect of the process, the need for sustainable and robust tools, the user experience in applying those tools as well as the involvement of the FAIR principles in this process.

Table of Contents

Inhalt

1. Introduction 7
2. Overview of the Ontology Engineering and Use Process 7
2.1 Ontology development methodologies 8
2.2 Common activities in ontology development processes 11
2.2.1 Ontology requirements specification 12
2.2.2 Ontology implementation 13
2.2.3 Ontology publication 13
2.2.4 Ontology maintenance 14
2.3 Ontology use and exploitation 14
3. Workshop on Ontology Engineering Tools 15
3.1 Questionnaire 16
3.2 Attendance 19
3.3 Presentations 20
3.4 Polls 22
3.5 Discussions 24
4. Specification of Ontology Ecosystem Toolkit 24
4.1 Components: Service requirements for ontology engineering and use 25
4.2 Additional requirements 26
5. Conclusion 27
References 27
Appendix A: Anonymised text from the chat facility of zoom during the workshop 30

List of Figures

Figure 1 Procedure for ontology design and evaluation 8
Figure 2 METHONTOLOGY development process and life cycle 9
Figure 3 On-To-Knowledge development process and life cycle 9
Figure 4 DILIGENT life cycle 10
Figure 5 NeOn scenarios for building ontology networks 11
Figure 6 LOT overview 12
Figure 7 Tool names and their frequency (for tools mentioned 3 times or more) extracted from responses to Question 3 17
Figure 8 Tasks and their frequency extracted from responses to Question 3 18
Figure 9 Categories of gaps identified from the responses to Question 4 19
Figure 10 Number of attendees having chosen each of the suggested categories (Question 1) 20
Figure 11 Tasks mentioned by respondents (with more than 2 responses) as the ones in which they are involved 20
Figure 12 Responses to the first poll sent during the workshop on other processes similar to ontology engineering 22
Figure 13 Responses to the second poll sent during the workshop on approach to be taken by a company getting into ontologies 23
Figure 14 Responses to the third poll sent during the workshop on features of importance for ontology engineering tools 24
Figure 15 Components of the ontology ecosystem toolkit 25

1. Introduction

One of the goals in OntoCommons is to provide a specification for the methodological and tool support to industry-targeted ontology development, maintenance and usage. In doing so, part of the work involves reviewing existing means to support the lifecycle of ontologies and identifying gaps and challenges in the development of ontologies. This is to be carried out in particular through engagement with relevant stakeholders, including ontology experts, practitioners, and users, as well as the developers of tools to support the ontology engineering and use process.

The objective of this report is to present the first iteration of the requirement engineering exercise carried out in this context in order to formulate the definition and extent of an ontology ecosystem from a methodological and technological point of views. This first specification includes a description of the components of the ontology ecosystem toolkit as well as other requirements that are associated with the realisation of this toolkit. Those components are derived mostly from three sources: Existing and adapted ontology engineering and use processes, represented in particular through the corresponding methodologies; learnings from the first focused workshop organised around the theme on ontology engineering tools; and the requirements gathered from the demonstrators considered in the project (see *report on requirements on ontology tools and ontologies and criteria for selection of further cases*).

We therefore here first describe the ontology engineering and use process (in Section 2), including a description of existing ontology engineering methodologies and of the main phases involved. In Section 3, we provide a description of the material collected through the focused workshop (and related activities) and an analysis of this material. Finally, in Section 4, we aggregate the learnings from those different sources to provide an overview of the components of the ontology ecosystem toolkit, as a set of services involved in the different phases of the ontology engineering and use process.

2. Overview of the Ontology Engineering and Use Process

The Ontology Engineering discipline investigates the principles, methods, and tools for initiating, developing, and maintaining ontologies. It provides life cycles that go from requirements definition to the maintenance of ontologies, as well as methodologies, techniques, and tools to support and drive the development of ontologies.

The following sections provide an overview of existing development methodologies, together with the common activities proposed by existing methodologies within the ontology engineering process and how such ontologies can be used and exploited. Considering the particularities of the OntoCommons use cases, this section considers as well the generation of Linked Data, APIs or SHACL

shapes for data validation, activities that are usually not included in the ontology development activities as they refer to their later exploitation and use.

2.1 Ontology development methodologies

This section includes a brief description of the five main traditional ontology development methodologies, based on the commons steps for ontology design and evaluation rather than present an extended review of methodologies[1], but focusing on those converging to a, to some extent, common development workflow. These methodologies are Grüninger & Fox (Grüninger et al., 1995), METHONTOLOGY (Fernández-López, 1997), On-To-Knowledge (Sure et al., 2004), DILIGENT (Pinto et al., 2009) and NeOn (Suárez-Figueroa et al., 2009). These methodologies are taken as background and their core components are reused in a novel methodology, the LOT methodology (see description below). The LOT methodology will be used and extended when necessary for the OntoCommons use cases ontology developments.

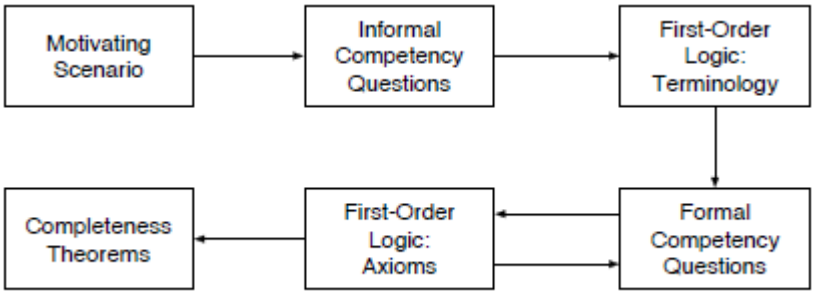

Figure 1 Procedure for ontology design and evaluation

Another existing methodology is METHONTOLOGY, which defines a set of life cycle models and a development process to provide an overview of how an ontology should be developed. This methodology identifies the set of activities to be carried out during the development process. The life cycle models that it proposed were the waterfall one, the incremental one (which ensures that each version is compatible with the previous ones) and the one based on evolved prototypes (with essential similarities to agile development). Figure 2 shows an overview of the ontology lifecycle and the activities proposed in this methodology.

From METHONTOLOGY the main technical activities as well as the evaluation and documentation activities are considered in the LOT methodology.

[1] A broader overview of existing methodologies will be included in the *report on Methodological framework for ontology management*

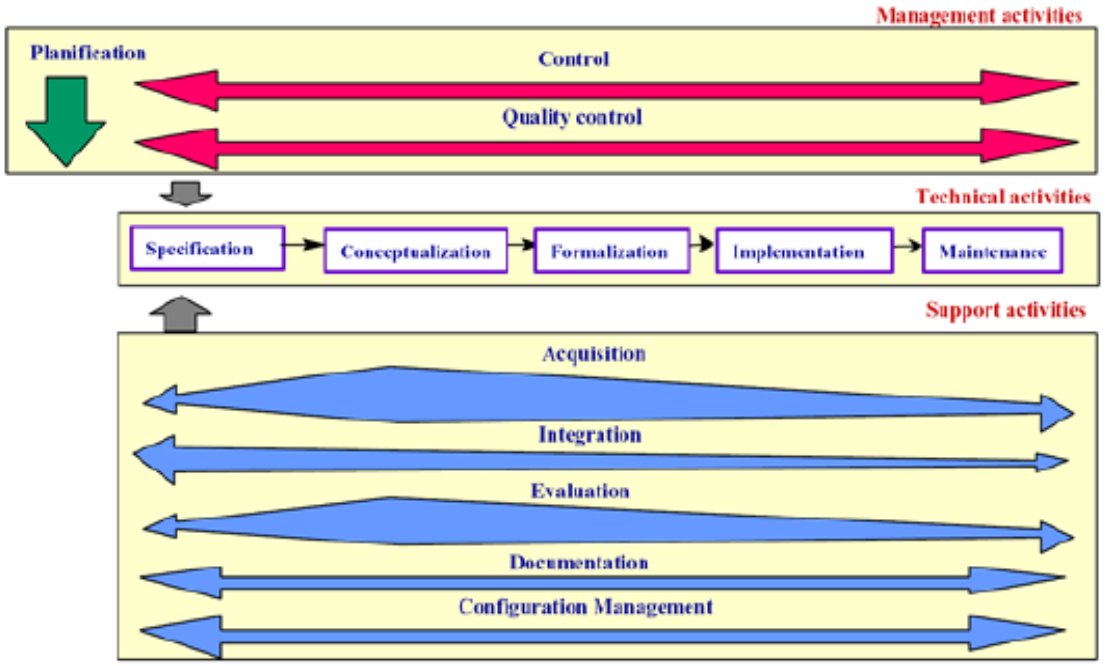

Figure 2 METHONTOLOGY development process and life cycle

The On-To-Knowledge methodology lies in the application-oriented development of ontologies. As it is shown in Figure 3, it includes five phases, namely: (a) Feasibility study, (b) Ontology kick-off, (c) Refinement, (d) Evaluation, and (e) Maintenance.

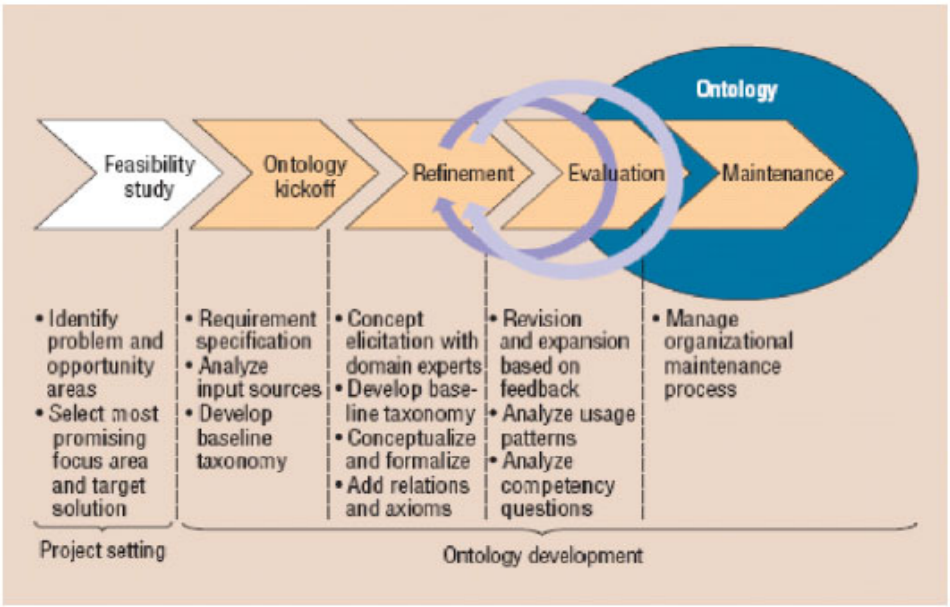

Figure 3 On-To-Knowledge development process and life cycle

During the ontology kick-off phase, ontology engineers should collect user requirements written as competency questions to provide an overview of possible queries to the system that can indicate the scope and content of the domain ontology. Next, during the evaluation phase, ontology engineers verify whether the target ontology satisfies the ontology requirements specification document and whether the ontology supports or answers the competency questions analysed in the kick-off phase of the project. During this evaluation phase new requirements that should be handled by the ontology can come up. The LOT methodology includes the refinement cycles in different phases inspired by this methodology.

Concerning the DILIGENT methodology, it was proposed to support ontology development in a distributed environment. In this scenario, the actors involved in the development of the same ontology have different complementary skills, including ontology users and ontology developers. The general process proposed in this methodology, shown in Figure 4 includes five activities, namely, (1) build, (2) local adaptation, (3) analysis, (4) revision and (5) local update. Technical consideration are included in LOT regarding the distributed edition of ontologies.

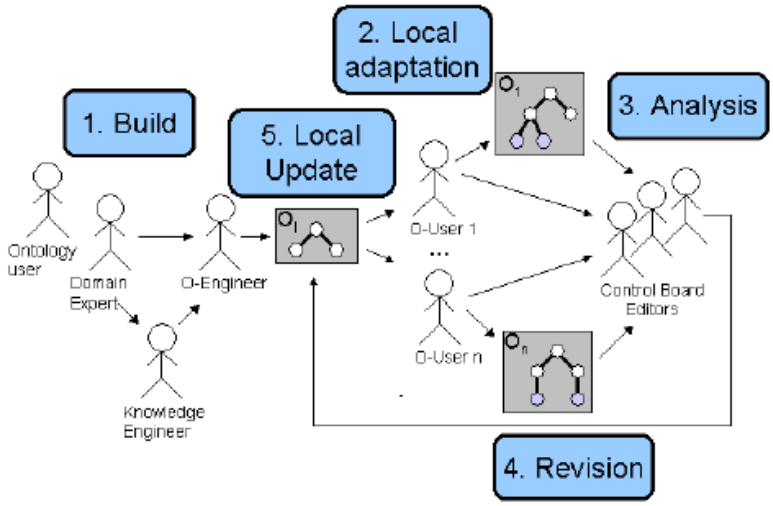

Figure 4 DILIGENT life cycle

Finally, the NeOn methodology was a combined European effort to provide precise guidelines, supported by an integrated development environment (the NeOn Toolkit[2]), to develop network ontologies. The main goal of these methodologies is to provide support for collaborative development of ontologies and for concrete guidelines for reuse and reengineering of knowledge sources. To do so, this methodology identifies nine scenarios for building ontology networks, which are the most common ones during ontology network development. An overview of all these scenarios is depicted in Figure 5.

[2] http://neon-toolkit.org/wiki

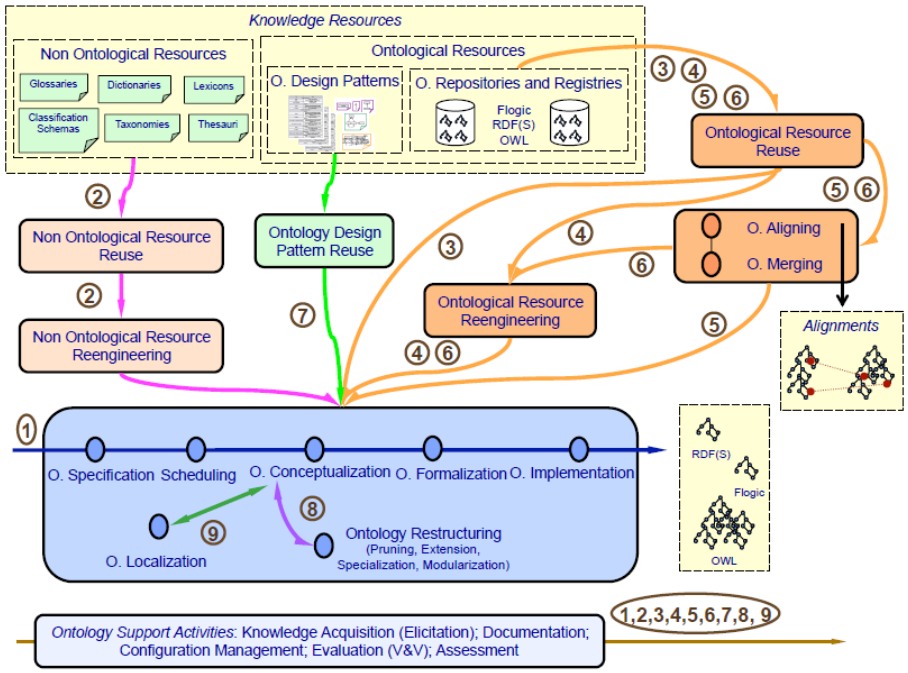

Figure 5 NeOn scenarios for building ontology networks

As shown in Figure 5 NeOn scenarios for building ontology networks, the methodology supports the ontology implementation activities, such as specification, conceptualisation, and formalisation, as well as the ontology support activities, such as the documentation and evaluation. The LOT methodology considers the reuse of ontologies and non-ontological resources reusing the NeOn methodology when applicable.

2.2 Common activities in ontology development processes

Current trends in Ontology Engineering show that lightweight and agile methodologies have become more popular (e.g., (Peroni, 2017) (Hristozova et al., 2002) (Presutti et al., 2009)) in the last years, where sprints and iterations represent the main workflow organization. Moreover, agile methodologies focused on the improvement of communication between ontology developers and domain experts, as well as the continuous assessment of the project status and the ability to respond to changes rapidly.

Taking as input the already mentioned existing methodologies, these current trends in Ontology Engineering and new needs that arose because of the Linked Data paradigm (e.g., the publication of ontologies), the ontology development process can be summarised into 4 activities, which are defined in the LOT methodology[3]: (1) Ontological requirements specification; (2) Ontology implementation; (3) Ontology publication; and (4) Ontology maintenance. It should be noted that previous methodologies did not consider nor oriented the development of the ontologies to their online publication following the best practices for publishing vocabularies and the compliance to the

[3] https://lot.linkeddata.es/

FAIR principles (see FAIR Semantics recommendations, P-Rec 4, Hugo et al., 2021). It should be noted that LOT intends to provide a general workflow covering and reusing state-of-the-art methodologies. Figure 6 shows an overview of these four activities, together with their outputs and inputs.

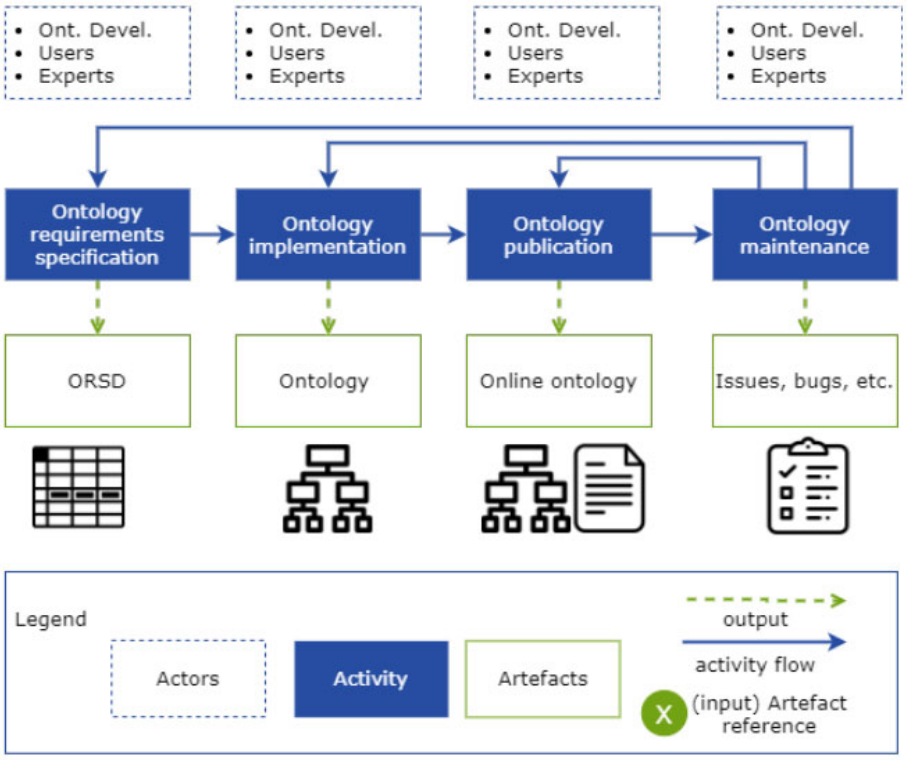

Figure 6 LOT overview

2.2.1 Ontology requirements specification

The *ontology requirements specification activity* aims to state why the ontology is being built and to identify and define the requirements the ontology should fulfil. In this step, involvement, and commitment by experts in the specific domain at hand is required to generate the appropriate industry perspective and knowledge.

Ontology requirements are divided into functional and non-functional requirements:

- Non-functional ontology requirements refer to the characteristics, qualities, or general aspects not related to the ontology content that the ontology should satisfy. Examples of non-functional requirements are: (a) whether the terminology to be used in the ontology must be taken from standards, (b) whether the ontology must be multilingual, or (c) whether the ontology should be written following a specific naming convention (e.g. following OBO Foundry principles or IOF principles).

- Functional ontology requirements, which can be also seen as content specific requirements, refer to the particular knowledge to be represented by the ontology, for example in the SEEMP case (Suárez-Figueroa et al., 2009), the knowledge about curriculum vitae with candidate skills, education level, expertise, previous work experience, or about job offers including job location, salary, etc. These functional requirements can be written as:
    - Competency questions, following the technique proposed by (Grüninger et al., 1995)
    - Natural language sentences
    - Tabular information, following the technique proposed by METHONTOLOGY (Fernández-López, 1997).

  If needed, functional requirements can be assigned to priorities, to allow planning and scheduling the development of the ontology in sprints.

2.2.2 Ontology implementation

The *ontology implementation activity* goal is to build the ontology using a formal language, based on the ontological requirements identified by the domain experts. In the LOT methodology the implementation activity is divided into the sub-activities: ontology conceptualization, ontology encoding, ontology reuse, and ontology evaluation. The ontology developers can schedule and plan the ontology development according to the prioritization of the requirements in the ontology requirements specification activity, if needed. The ontology development team builds the ontology iteratively, implementing a certain number of requirements in each iteration. The output of each iteration is a new version of the ontology.

First, during the ontology conceptualization, the model that represents the domain of the ontology is created, although it does not include all the constraints that a formal language imposes. Afterwards, when the model is ready, the ontology is encoded using an implementation language, e.g., OWL. Both before or after the ontology conceptualization, developers can reuse existing ontologies. Finally, once the ontology is implemented, it must be evaluated to check that there are no modelling mistakes and that all the requirements are satisfied.

2.2.3 Ontology publication

In the LOT methodology this essential activity is divided into the following sub-activities: proposed release candidate, ontology documentation and online publication. During the *ontology publication activity,* the ontology development team should provide an online ontology which is accessible both as a human-readable document, i.e., as an HTML document with the description of the ontology and its terms, and a machine-readable file from its URI therefore enforcing FAIR Principles.

Such HTML documents should include a human-readable description of the ontology that describes the classes, properties, data properties and individuals of the ontology. Additionally, the HTML description of an ontology should contain metadata, such as the license URI, the title being used, the

creator, the publisher, the date of creation, the last modification or the version number, and it should also contain information oriented to human consumption about the intended use of the ontology, its purpose and scope. Finally, diagrams that store the graphical representation of the ontology, as well as example of use are also useful for the readability of the ontology.

2.2.4 Ontology maintenance

Finally, the goal of the *ontology maintenance activity* is to update the ontology. This may be needed due to different situations, such as new requirements identification, bugs detection or improvement suggestions. This activity may be triggered during the ontology development process in which the new requirements implementation or bug fixing would be scheduled in one or more sprints. This activity may be also triggered after the ontology development process in which a new version or revision of the ontology should be generated.

2.3 Ontology use and exploitation

Among other purposes, ontologies are commonly used for transforming data into a structured RDF format. Following Radulovic and colleagues work (Radulovic et al., 2015), the first step of this data transformation is to select the RDF serialization (RDF/XML, Turtle, N-Triples and JSON-LD). Afterwards, it is needed to select a tool for data transformation, depending on the format of the data (database, spreadsheets, etc.) and on concrete needs of the transformation process (e.g., dynamicity, scalability). The data transformation usually requires defining a mapping between the data and the ontology. Such mapping could be conceptual or explicit using mapping languages as RML. Finally, it is needed to evaluate different aspects of the obtained RDF data set. For example, checking for literals that are incompatible with the data type ranges, checking whether the data set contains redundant objects, checking whether the data set uses existing established ontologies to represent its entities, or determining whether a data set provides possibilities for obtaining the necessary information (e.g., in terms of SPARQL queries).

There are tools available for transforming data from databases (e.g., morph-RDB (Priyatna et al., 2014), D2R Server[4], TopBraid Composer, Ontop[5]), XML files (e.g., OpenRefine[6], TopBraid Composer, Karma[7]) or spreadsheets (e.g., Excel2rdf[8], XLWrap[9], TopBraid Composer, OpenRefine). Once the data is transformed into RDF it can be made available for user in different ways, for example by providing SPARQL endpoints.

[4] http://d2rq.org/d2r-server

[5] https://ontop-vkg.org

[6] https://openrefine.org/

[7] https://usc-isi-i2.github.io/karma/

[8] https://pypi.org/project/excel2rdf/

[9] https://xlwrap.sourceforge.io/

Another use of ontologies is the generation of shapes e.g., SHACL[10], which is a W3C recommendation, or ShEx[11] shapes. A shape defines a set of restrictions that data must fulfil, which can be restrictions related to the data model (e.g., cardinality) and restrictions that apply to the data values (e.g., string patterns). These shapes offer the possibility to validate RDF graphs against a set of conditions and to implement closed world assumptions. Moreover, shapes can also be considered as a description of the data graphs that do satisfy these conditions. Such descriptions may be used for not only for validation, but also for code generation and data integration.

Finally, APIs can also be generated from ontologies to help developers to consume data. Such APIs can be generated from the competency questions and the ontology code (ontology artefacts generated during the ontology development process). As an example, given the ontology code http://vocab.linkeddata.es/vgo and the competency question "What games has the player played?" the associated API can have the path "/players/:id/games" which retrieves the list of games.

The LOT methodology presented here is a framework from which we will be able to extract specific requirements for the Ontology Engineering Toolset. The LOT methodology has been developed by reusing well-known and broadly adopted practices in ontological engineering and including specific activities for the publication of the resulting ontologies following the semantic web best practices. In addition this methodology is inspired by software common practices to couple the ontology development to software project considering ontologies first class citizens in any software or application development.

3. Workshop on Ontology Engineering Tools

In this section, we described the online workshop organised on the 19th of March under the title "OntoCommons Workshop on Tools for Ontology Engineering".[12] The objective of the workshop was to collect from the community of ontology practitioners, tool developers and users of ontologies information about the state of existing ontology ecosystem toolkits, about their usage and about the gaps that was perceived in toolkits supporting the different steps of the ontology engineering and usage tasks as described above in Section 2.

The workshop was described in the following way:

It might be said, similarly to many other roles and activities that an ontology engineer is only as good as his/her tool, but what are the tools for ontology engineering and how good are they? The OntoCommons Workshop on Tools for Ontology Engineering is an opportunity to discuss among ontology engineering experts, practitioners, tool builders and users about the state of the ontology engineering toolkit in 2021. The event aims to cover the whole ontology lifecycle, from requirement

[10] https://www.w3.org/TR/shacl/

[11] https://shex.io/

[12] https://ontocommons.eu/news-events/events/ontocommons-workshop-tools-ontology-engineering#:~:text=The%20OntoCommons%20Workshop%20on%20Tools,ontology%20engineering%20toolkit%20in%202021.

gathering to ontology use through a mix of presentations by diverse members of the ontology engineering community and discussions to understand which tools contribute to which parts of the ontology lifecycle and what is currently lacking. These presentations and discussions will be used as input to a report from the OntoCommons project on the state of ontology engineering tools in 2021.

Specific invitations were sent to known members of the community, and a broader open invitation was sent at a later stage, once the list of speakers was finalised. The registration form included a questionnaire, described in Section 3.1 below, regarding the relationship of the respondents with ontology engineering and their view of the gaps that existed in tool support for ontology engineering and use. We received 132 responses to this questionnaire, corresponding to 125 unique registrations, (it was possible to respond and not register, see Section 3.2 for the details of the questionnaire).

The programme of the workshop included, besides an introduction to the project and the workshop, 11 presentations and a panel session (see Section 3.3). Discussions were collected through the chat facility of zoom and notes were taken by OntoCommons members on each of the presentations. Four small polls were also conducted during the workshop using the sli.do tool (see Section 3.4).

3.1 Questionnaire

The questionnaire included with the registration form included four different questions:

- With respect to ontology engineering, do you consider yourself (multiple choice question with options such as "An expert practitioner" and an "Other" option).
- In which part(s) of the ontology engineering process are you generally (would you like to be) involved? (multiple choice question including options such as "Ontology authoring/editing" and an "Other" option).
- Please list tools that you use (would like to use) in the ontology engineering workflow. For each tool, please indicate the part(s) of the process it mostly supports (open text question).
- Please list functions and features that you think are missing from those tools, or that are not sufficiently supported by those tools in your experience (open text question).

The two first questions are considered in the attendance section below. Regarding Question 3, Figure 7 below shows the subset of the 85 tools extracted from the responses that were mentioned by at least 3 respondents. As can be seen, the tool that is by far the most known and used is Protégé, which is at its core an ontology editor. Interestingly, the tools ranking second and third are also mostly used for ontology authoring and editing. It is surprising to see however that, next to two very complex ontology editors (Protégé and TopBraid Composer), text editors are the most used tools for this task. Other tools for ontology edition are also mentioned, but at much lower ranks.

Next, we find a number of tools and libraries mostly used for storing and querying RDF data and ontologies, including GraphDB, Jena, OWL API, Virtuoso, rdf4j, Fuseki and rdflib. Those are often mentioned in relation to the task of ontology-based application development, but can also be used for querying as part of the core ontology engineering process.

Regarding ontology drafting, concept identification and requirement elicitation, only a few tools are mentioned to support those tasks, namely Cameo and OWBO. Generic graph editors are also cited for this task.

Ontology validation is a commonly mentioned task, for which SHACL is often mentioned. OOPS! is also used for this task.

Regarding publication and deployment, Widoco is the only tool for ontology documentation that received more than 3 mentions, but it is interesting to see GitHub, a general version control and issue tracking system, being mentioned here. Very few ontology repositories for publication and reuse are ever mentioned, and LOV (linked open vocabularies) is the only one making more than three appearances.

Several reasoners also make an appearance with Hermit and Pellet being the most commonly mentioned.

Finally, in terms of visualisation and visual editing, while tools like gra.fo were mentioned, WebVOWL is the only one that appears to be sufficiently known by the community to appear in the chart.

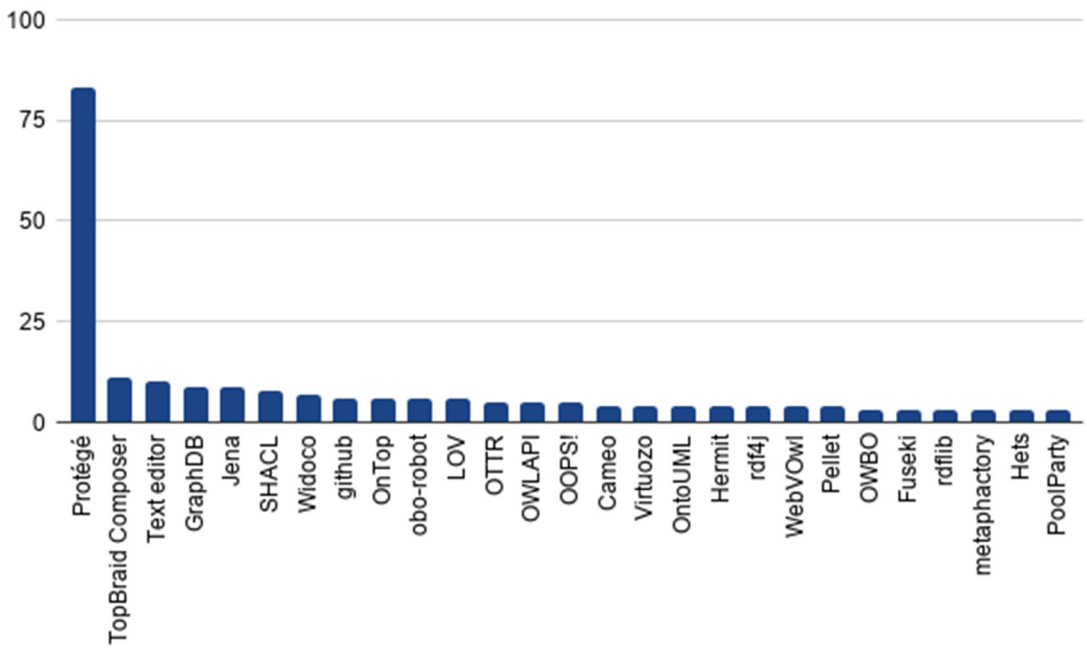

Figure 7 Tool names and their frequency (for tools mentioned 3 times or more) extracted from responses to Question 3

In Question 3, respondents were also asked to indicate the tasks for which they were using those tools. Figure 8 below shows the tasks extracted from the responses and their frequency. Unsurprisingly, ontology editing and authoring is the most commonly considered task. However, considering the low number of tools mentioned that have for core feature to support it, it is more surprising that ontology drafting and concept identification received a large number of mentions. A similar remark can be made of the population, visualisation, reuse, and requirement elicitation tasks.

Those are currently, it appears, being carried out mostly with tools that are not necessarily designed for them. Surprisingly as well, while no specific tool is mentioned for those tasks, modularisation and ontology matching are mentioned multiple times as tasks carried out with existing tools.

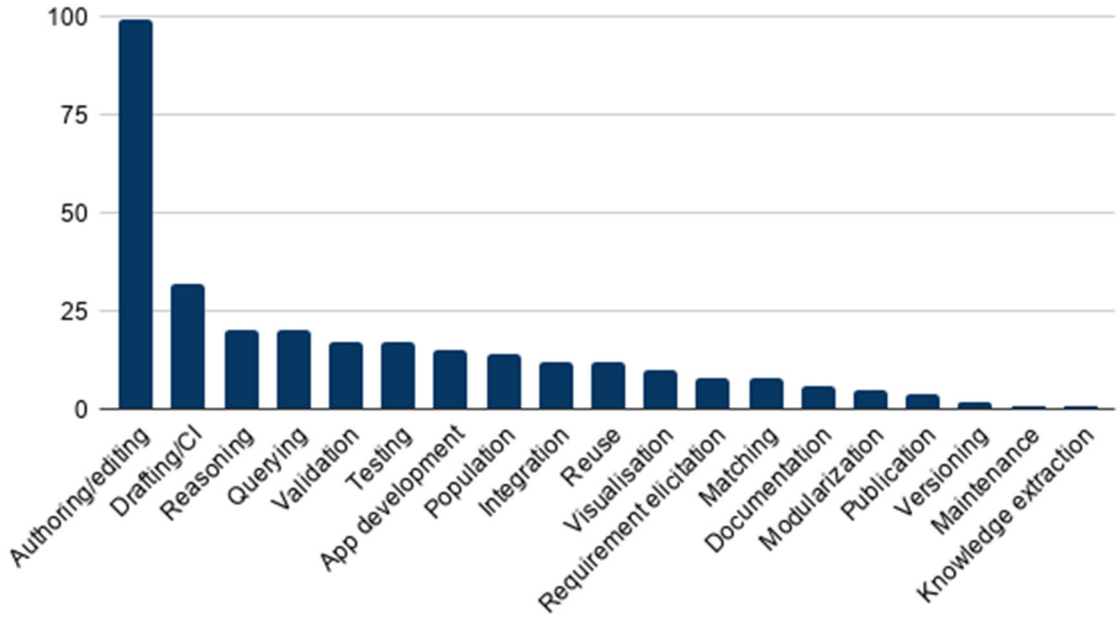

Figure 8 Tasks and their frequency extracted from responses to Question 3

Regarding Question 4, concerning the gaps in existing tool support, Figure 9 below shows the categories of answers extracted from the responses and their frequencies. Visualization makes a strong appearance, which is consistent with the low number of tools supporting it. Even though tools such as the ones based on SHACL are well established, ontology validation appears to still be a problematic area. In addition, many criticisms related to the tools not being sufficiently integrated with each other, offering poor user experience, and not enabling sufficient expressiveness.

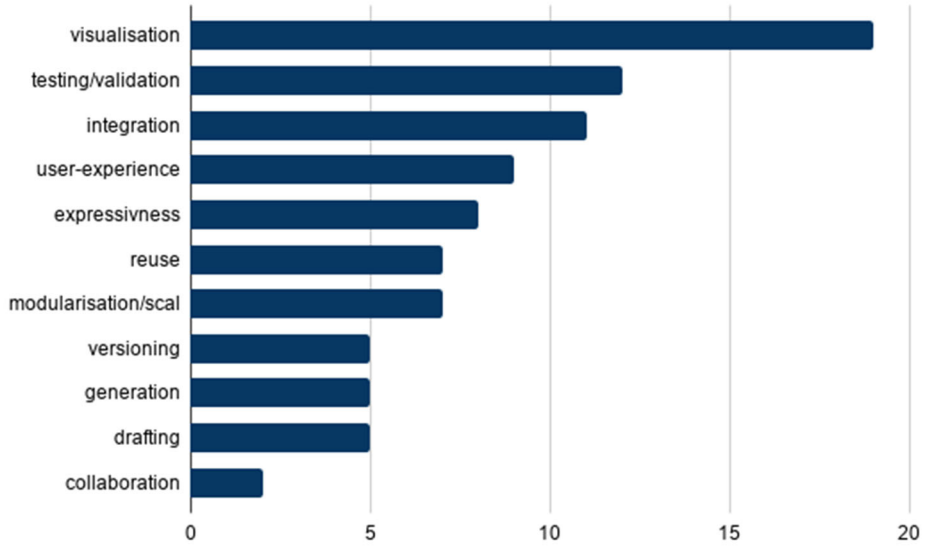

Figure 9 Categories of gaps identified from the responses to Question 4

3.2 Attendance

As mentioned earlier, we received 132 responses to the questionnaire, corresponding to 125 unique registrations to the workshop. The workshop took place between 10am and 4pm GMT. While there was some variation during the day, with some attendees coming in and out, there were always more than 80 attendees connected to the workshop, and the number of attendees peaked at 103.

Figure 10 below shows the number of respondents having chosen the categories suggested in Question 1 of the questionnaire (with more answers including other categories). As can be seen, there were close to the same number of experts, practitioners and tool builders, with many attendees falling in more than one category. The "Someone who would like to be more involved", representing the more novice attendees, has been less used by respondents. This was expected since the workshop was intended for the ontology engineering community and invitations were made specifically to target the three first categories.

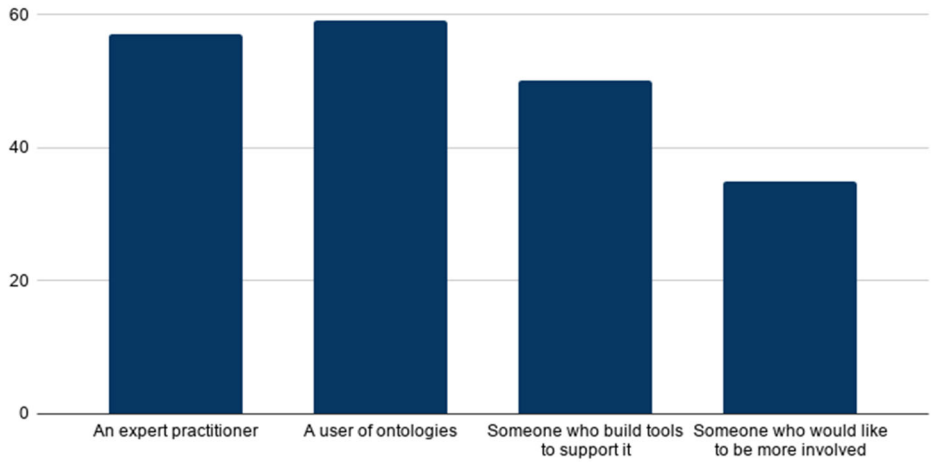

Figure 10 Number of attendees having chosen each of the suggested categories (Question 1)

Figure 11 below shows the number of respondents having chosen each of the possible tasks included in the ontology engineering process. Unsurprisingly, the most commonly considered tasks are the ones that are core to the ontology building and implementation process: Concept identification, ontology drafting, ontology authoring/editing, ontology population. Some of the more optional tasks in this process, such as ontology modularisation and alignment are however amongst the least represented. Interesting, considering the two tasks that related the use of ontologies, application development and ontology querying, the former is the most cited while the later is amongst the least cited.

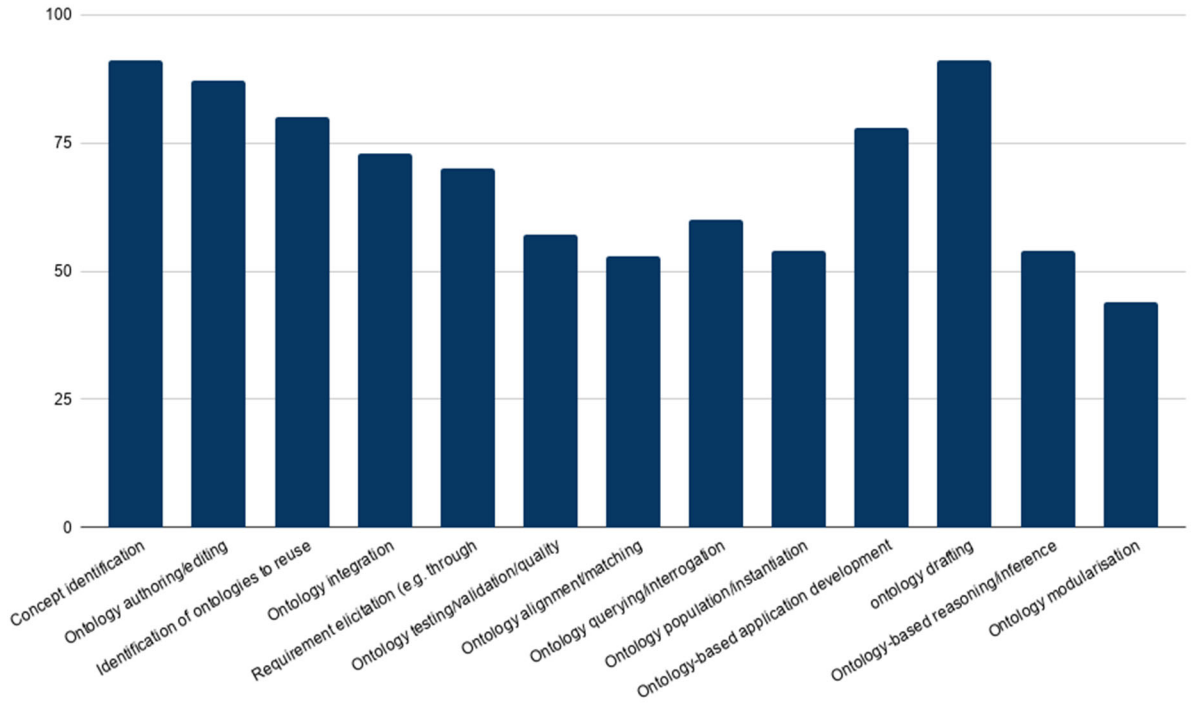

Figure 11 Tasks mentioned by respondents (with more than 2 responses) as the ones in which they are involved

3.3 Presentations

The Schedule of the workshop is included below. There were 11 presentations in total in three different sessions.[13] The presentations were delivered through the zoom tool, with Q&A slots at the end of each session. During the presentations, attendees were discussing through the chat facility of zoom and responded to polls. At the end of the day, a panel was organised, with each of the panellists presenting a short statement on the state and issues of ontology engineering tools in 2021. Notes were taken by OntoCommons members during the presentations on a shared online document.

[13] The first presentation had to be cancelled as the speaker fell ill on the day.

10am	Introduction	Mathieu d'Aquin
10.15am	The Fashion Knowledge Graph at Zalando	Katariina Kari
10.30am	From regulation to knowledge	Veronika Heimsbakk
10.45am	Ontology Engineering at eccenca - Overview and Outlook	Sebastian Tramp
11am	Ontology repositories & services with the OntoPortal technology: a return of experience with AgroPortal	Clement Jonquet
11.15am	Q&A	
11.30am	Lunch break	
12.45pm	Conceptual Modelling with ORMiE in Microsoft Visual Studio	Enrico Franconi
1pm	UFO, OntoUML and its Associated Tools	Giancarlo Guizzardi
1.15pm	Building the ArCo knowledge graph: process, experience and struggle with existing tools	Valentina Presutti
1.30pm	Visual Ontology Modeling for Domain Experts and Business Users with metaphactory	Peter Haase
1.45pm	Q&A	
2pm	Semantic Requirements in the Ontology Lifecycle	Michael Gruninger
2.15pm	DOL and Hets - Tools for modular and heterogeneous ontologies	Till Mossakowski
2.30pm	Ontology Engineering for Developer Ergonomics: the data.world use case	Juan Sequeda
2.45pm	Q&A	
3pm	Panel on the suitability of current tools for ontology engineering: Is there a standard approach? Are we missing something? Are the tools good enough?	Mehwish Alan Enrico Motta Aldo Gangemi Maria Keet Stephen Kahmann
3.30pm	Wrap up	

Some of the presentations, such as the one given by Veronika Heimsbakk, focused on use cases and scenarios regarding the building and usage of ontologies. In those presentations, much of the focus was on the aspects of modelling ontologies according to domain/application requirements and validating them (e.g. through SHACL).

While, as discussed in Section 2, ontology reuse is very important in ontology engineering, surprisingly only one of the presentations, the one given by Clement Jonquet (as well as some discussions in the panel), focused on this aspect through presenting ontology repositories. It should be mentioned however that an ontology repository was also used as a way of validation/modelling in Michael Gruninger's presentation.

Besides those, most of the other presentations focused on specific tools dedicated to modelling and developing ontologies, including OntoUML (Giancarlo Guizzardi), gra.fo (Juan Sequeda) and metaphactory (Peter Haase).

The panellist presented their views of issues with tool support for ontology engineering, including different aspects. This will be further described in Section 3.5.

3.4 Polls

Sli.do is an online tool to collect questions and create short, simple polls during presentations. During the workshop four short polls were shared with participants to collect basic information and opinions about aspects of the ontology engineering process.

In the first poll, the question asked was "In your view, the ontology engineering process is most similar to:". Figure 12 below shows the results from 41 responses, indicating that the majority of respondents would consider it to be more similar to the software engineering process.

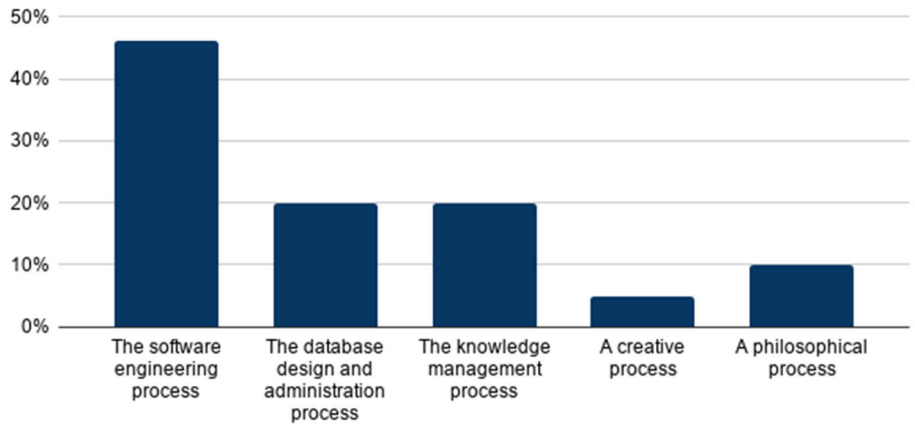

Figure 12 Responses to the first poll sent during the workshop on other processes similar to ontology engineering

In the second poll, the question asked was "In your opinion, how well do tools support the ontology engineering process across the whole lifecycle of ontologies?". This was a rating poll, with the average over 28 responses being 2.3 out of five stars.

In the third poll, the question asked was "In your view, a company wanting to develop and use ontologies should:", with multiple choices to select. The results are very clear that most of the 41 respondents believed that hiring an ontologist would be the best approach, as shown in Figure 13.

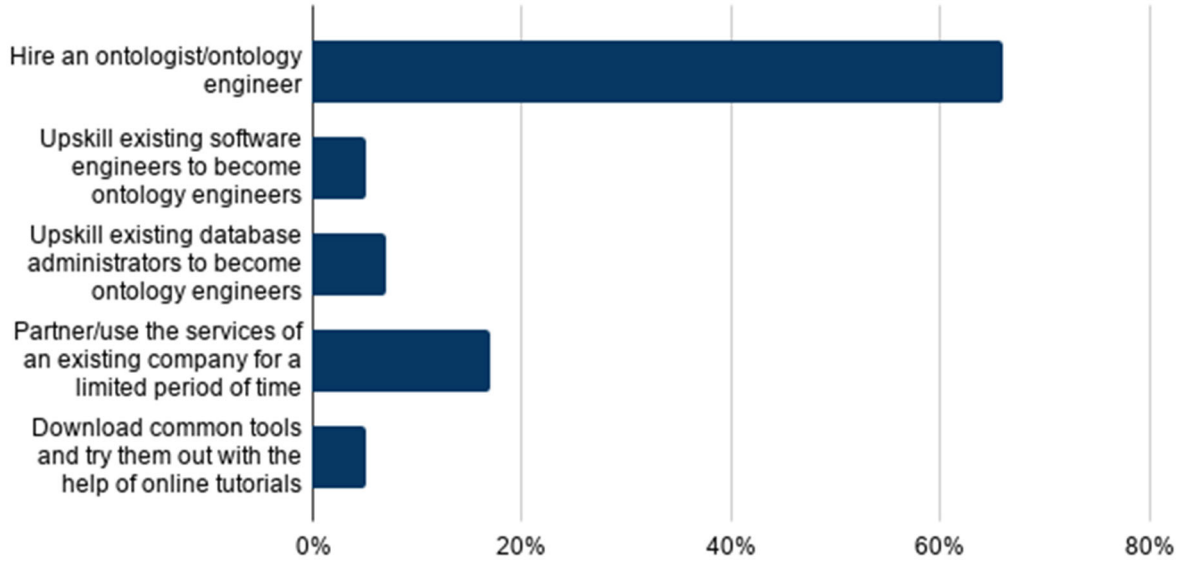

Figure 13 Responses to the second poll sent during the workshop on approach to be taken by a company getting into ontologies

In the fourth poll, the question asked was "What do you most expect from an ontology engineering tool?". This was a ranking poll, which is one where respondents had to put the different options in an order of importance. As can be seen in Figure 14 below, ontology editing and authoring are confirmed as the core feature of an ontology engineering tool. Other features such as reasoning, and initial conceptualisation are scored close to each other. Interestingly, the last step of the ontology engineering process before usage, deployment and publication, is the least considered feature.

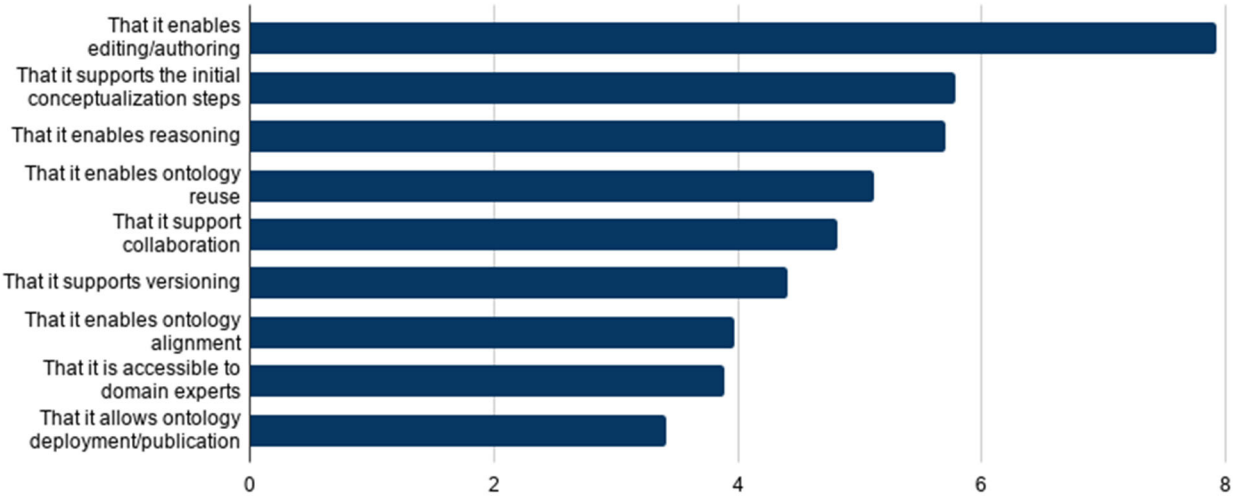

Figure 14 Responses to the third poll sent during the workshop on features of importance for ontology engineering tools

3.5 Discussions

Appendix A of this document includes an anonymised version of the discussions that happened through the chat facility of the zoom tool during the workshop. As can be seen, considering the online format of the workshop, those discussions were very active and constantly happening during the day. Also, many of the same tools and gaps already identified are included in those discussions. It is however also interesting to see that other topics tend to have prominence in the chat discussions. Those include in particular aspects related to the cost associated with the use of ontology engineering tools. The related question of the sustainability and robustness of those tools is also brought up, since many are open source software developed as part of funded projects. The aspect of the usability and of the user experience offered by ontology engineering tools appear more critical through the chat discussion than through the presentations and the questionnaire. It was also a key topic of the statements and discussions during the panel, with panellist making the argument that those tools offer a poor user experience, which is hindering their adoption and sustainability. Other topics addressed during the workshop is the lack of integration between ontology engineering tools, the need for better support for collaborative development in ontology engineering and the need for better support for validation in particular in the context of collaboration with domain experts. Those topics are consistent with the ones already mentioned in the questionnaire, as described in Section 3.1

4. Specification of Ontology Ecosystem Toolkit

In this section, we rely on the input and learnings from the workshop as summarised above as well as from other sources, including the requirements of OntoCommons demonstrators as described in the *report on requirements on ontology tools and ontologies and criteria for selection of further cases*, to identify first a set of components for the Ontology Ecosystem Toolkit of the project, as well as other (mostly non-functional) requirements for this toolkit.

4.1 Components: Service requirements for ontology engineering and use

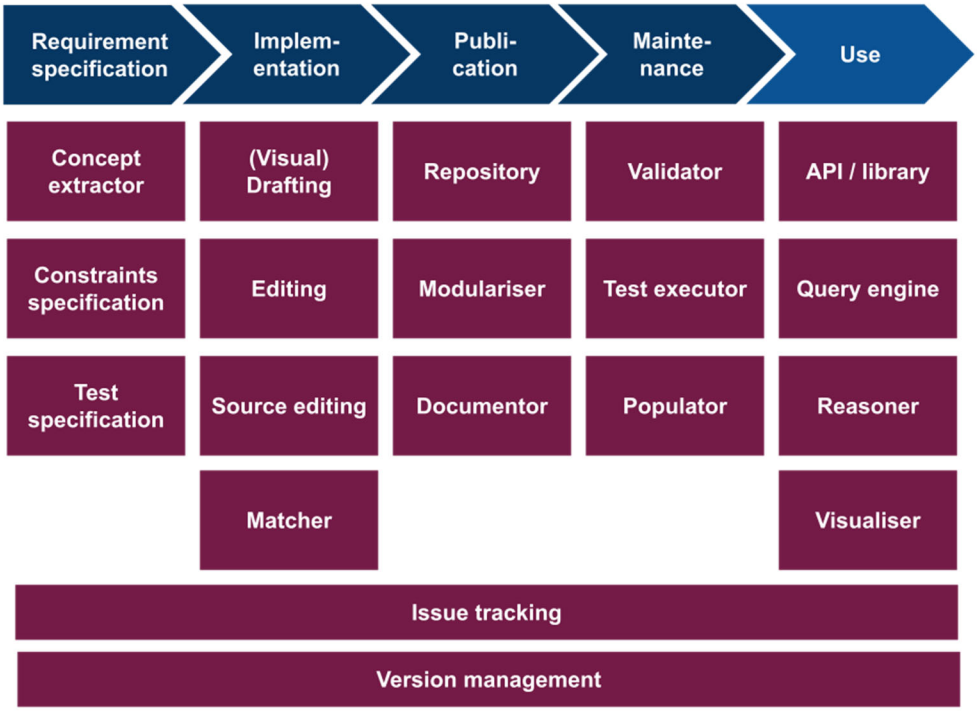

Figure 15 Components of the ontology ecosystem toolkit

Figure 15 above provides an overview of the components of the ontology ecosystem toolkit supporting ontology engineering and use nowadays, and on which the OntoCommons project is to rely. Those components are loosely classified under the four steps of the ontology engineering process and the "ontology use and application" step described in Section 2 of this document. However, it is obvious that those steps are rarely applied sequentially in the exact order presented here, and that a number of those components can have a role in several of those steps. The placement of those components under those steps is therefore purely for the purpose of structuring the figure based on which phase of the process a component subjectively mostly contributes to.

While some of those components can be seen as obvious and would have naturally appeared in other similar specifications in the past, others are worth emphasising. Those include the concept identification, constraint specification, test specification and visual drafting components which have

been consistently identified during the workshop and in other occasions as lacking tool support at the moment. The aspects of validation and test execution that follow are therefore also worth highlighting, as is the component related to populating and instantiating the ontology which can support a number of "subtasks" such as connecting the ontology to a relational database in which related data is held.

Each of the included components is seen as a service in a broad sense, i.e. a piece of software or a part of a broader tool that provides a specific function within the broader process of ontology engineering and use. As such, they are expected to interact with each other, and some dependencies exist. A clear example of this is the relation between the constraint specification service and the validation service. Other implicit connections exist in this diagram as well between the ontology authoring service and a number of "peripheral" services for example related to modularisation or ontology matching.

Finally, three of those components should be mentioned for having a broader role. The issue tracking and versioning management services in particular take a similar role as in a software engineering process, across all the steps. In addition, while placed under the publication step, an ontology repository service plays a broader role as it might be involved in drafting and authoring for the purpose of ontology reuse.

4.2 Additional requirements

In addition to the functions represented by the components above, a number of mostly non-functional requirements can be identified. One of the key one relates to the architecture of the ontology ecosystem toolkit: On the one hand, it has been identified that integration between the different tools is critical, but on the other hand, it is important for it to be considered an open toolkit that can flexibility integrate different tools. Here, it is therefore important for the toolkit to exist as a set of interoperable services. Future versions of this specification should therefore focus on establishing interoperable interfaces for those services, ensuring in this way that they are both integrated and flexible.

A key aspect mentioned several times during the workshop and as part of requirement gathering exercises for the OntoCommons demonstrators is the importance of the user experience in those tools. In selecting tools to implement each of the considered services, particular attention needs to be given to the robustness and interaction aspects of those tools. In addition, some of those components, especially those involved earlier in the process, should be specifically targeted at enabling the involvement of domain experts in the development of ontologies.

In relation to this last point, a requirement expressed in multiple sources is also that the process of ontology engineering and use should be collaborative. Multiple users should be able to contribute to each phase of the process, which requires the services to be implemented in a way that enables such collaborations to happen.

It is useful here also not to lose sight of the fact that an important requirement is for the ontologies produced/published following the recommended process and through the components identified

above should follow the FAIR principles (Findable, Accessible, Interoperable, and Reusable). A number of those components are included specifically for this purpose such as the repository service or the documentor, but the particular implementation of those services needs to be carried out in a way that follows those principles and take into account various aspects that were not discussed during this workshop such as metadata description of the ontologies, the use of Globally Unique Persistent and Resolvable Identifiers, licensing, alignment with Top Level ontologies, etc. These specific requirements for tools will be extracted from the ongoing work done in the context of the FAIRsFAIR (FAIR Semantics recommendations, Hugo. et al, 2021) project to provide FAIR enabling tools during the ontology engineering process. These requirements should mostly affect ontology editors which are the entry point for most of the ontologists as shown by the survey results but also the documentor and repository tools.

Finally, this report, while also introducing methodological aspects, mostly focuses on the toolkit within the Ontology Ecosystem. It is important to note that such a toolkit is only one part of the ecosystem, and that other parts should affect it and be affected by it. As already mentioned, the demonstrators considered in the project are already strongly linked to the requirements and components considered here. The ontologies and communities involved in this ecosystem also need to be considered.

5. Conclusion

In this document, we reported on the first focused workshop organised as part of work package 4 of the OntoCommons project, and on how this and other sources contributed to the first version of the specification of the ontology ecosystem toolkit. This specification takes the form of a set of components, considered as services supporting the tasks of the ontology engineering and use process as well as additional, mostly non-functional requirements.

As mentioned earlier in this document, this represents the first iteration in this specification. The exercise of defining the components and requirements for the ontology ecosystem toolkit also strongly relate to further tasks of the project. Those include in particular the landscape analysis, which will build on some aspects presented here to provide a registry of tools for ontology engineering and use. Also, strongly related is the work on the methodological aspects where the elements presented in Section 2 will be further refined and aligned with the objectives and requirements of the project.

6. References

A Simplified Agile Methodology for Ontology Development [Conferencia] / aut. Peroni Silvio // OWL: Experiences and Directions – Reasoner Evaluation. - Bologna, Italy : [s.n.], 2017.

An eXtreme method for developing lightweight ontologies [Conferencia] / aut. Hristozova Maia y Sterling L.. - 2002.

Astrea: Automatic Generation of SHACL Shapes from Ontologies [Conferencia] / aut. Cimmino Andrea, Fernández-Izquierdo Alba y García-Castro Raúl // Extended Semantic Web Conference. - 2020.

Automating Ontology Engineering Support Activities with OnToology [Publicación periódica] / aut. Alobaid Ahmad [y otros] // Journal of Web Semantics. - 2019. - Vol. 57.

eXtreme Design with Content Ontology Design [Conferencia] / aut. Presutti Valentina [y otros] // Proceedings of the 2009 International Conference on Ontology Patterns. - Washington DC : [s.n.], 2009. - Vol. 516.

Formalisation and Experiences of R2RML-Based SPARQL to SQL Query Translation Using Morph [Conferencia] / aut. Priyatna Freddy, Corcho Oscar y Sequeda Juan // International conference on World Wide Web . - Seoul : [s.n.], 2014.

Methodology for the Design and Evaluation of Ontologie [Conferencia] / aut. Grüninger Michael y Fox Mark S.. - 1995.

Methontology: from ontological art towards ontological engineering. [Publicación periódica] / aut. Fernández-López M., Gómez-Pérez, A., & Juristo, N. - 1997.

NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology [Conferencia] / aut. Suárez-Figueroa Mari Carmen y Gómez-Pérez Asunción // Proceedings of the International Conference on SOFTWARE, SERVICES & SEMANTIC TECHNOLOGIES. - Sofia, Bulgaria : [s.n.], 2009. - págs. 160-167.

NeOn Modelling Components. Deliverable 5.1.1 NeOn Project [Informe] / aut. Suarez-Figueroa Mari Carmen [y otros]. - 2007.

OntoCheck: verifying ontology naming conventions and metadata completeness in Protégé 4 [Publicación periódica] / aut. Schober Daniel [y otros] // Journal of Biomedical Semantics. - 2012. - 2 : Vol. 3.

On-To-Knowledge Methodology [Sección de libro] / aut. Sure York, Staab Steffen y Studer Rudi // Handbook on Ontologies / aut. libro Staab Steffen y Studer Rudi. - [s.l.] : Springer Berlin Heidelberg, 2004.

On-to-knowledge methodology (OTKM). [Sección de libro] / aut. Sure Y., Staab, S., & Studer, R. // Handbook on ontologies. - [s.l.] : Springer Berlin Heidelberg., 2004.

Ontologies: Principles, methods and applications. [Publicación periódica] / aut. Uschold Mike y Gruninger Michael // The Knowledge Engineering Review. - 1996. - 2 : Vol. 11.

Ontology engineering and evolution in a distributed world using DILIGENT [Sección de libro] / aut. Pinto H Sofia, Tempich Christoph y Staab Steffen // Handbook on ontologies. - 2009.

OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation [Publicación periódica] / aut. Poveda-Villalón María, Gómez-Pérez Asunción y Suárez-Figueroa Mari Carmen // International Journal on Semantic Web and Information System. - 2014.

The DILIGENT knowledge processes [Publicación periódica] / aut. Vrandečić Denny [y otros] // Journal of Knowledge Management. - 2005.

The Live OWL Documentation Environment: A Tool for the Automatic Generation of Ontology Documentation [Conferencia] / aut. Peroni Silvio, Shotton David y Vitali Fabio // Internation conference on Knowledge Engineering and Knowledge Management. - Galway : [s.n.], 2012.

The TDDonto tool for test-driven development of DL knowledge bases [Conferencia] / aut. Lawrynowicz Agnieszka y Keet C Maria // Description Logics. - 2016.

Themis: a tool for validating ontologies through requirements [Conferencia] / aut. Fernández-Izquierdo Alba y García-Castro Raúl // International Conference on Software Engineering and Knowledge Engineering. - Lisbon, Portugal : [s.n.], 2019.

Guidelines for Linked Data generation and publication: An example in building energy consumption [Publicación periódica] / aut. Radulovic Filip [y otros] // Automation in Construction. - 2015. - Vol. 57.

VoCol: An Integrated Environment to Support Version-Controlled Vocabulary Development [Conferencia] / aut. Halilaj Lavdim [y otros] // International conference on Knowledge Engineering and Knowledge Management. - Bologna : [s.n.], 2016.

WIDOCO: A Wizard for Documenting Ontologies [Conferencia] / aut. Garijo Daniel // International Semantic Web Conference. - Vienna : [s.n.], 2017.

D2.5 FAIR Semantics Recommendations / Hugo, Wim, Le Franc, Yann, Coen, Gerard, Parland-von Essen, Jessica, & Bonino, Luiz. (2020). Second Iteration (Version 1.0). DOI : 10.5281/zenodo.4314321

How to write and use the ontology requirements specification document / Suárez-Figueroa, Mari Carmen, Asunción Gómez-Pérez, & Boris Villazón-Terrazas. OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (2009) Springer, Berlin, Heidelberg.

Appendix A: Anonymised text from the chat facility of zoom during the workshop

Link is provided to Slido to allow participants to interact live in the workshop.

==========

Protégé (https://Protégé.stanford.edu/) appears to not be properly supported anymore, with just one commit on the GitHub project page in more than a year and a lot of issues remain "not cared". A suggestion is made to move to a different platform.

The focus for Protégé has shifted to the web version, and the mailing list is very active. As Protégé is open source it is suggested that we can (as a community) jump in. The web version has basic functionalities but is not as complete as the desktop and does not have the same plugins. It has no reasoner and no control over ontology identifiers. However, there does not appear to be an open source alternative.

TopBraid Composer is often cited as an alternative although there is a discussion as to whether academic licences have been discontinued. It had been withdrawn for a period, but is now available again for two years at $695 which is seen to be fair. In addition, there is also a free version (https://www.topquadrant.com/topbraid-composer-install/) that covers basic functionality. It is felt by some that it does not include many important functions, but others feel that it is possible to use for real-world projects, though it lacks automation and integration functionalities. The commercial version is expensive.

One partner develops an open-source platform called Mobi (https://github.com/inovexcorp/mobi) that may be of interest. It takes a different approach than a tool like Protégé and focuses more on collaborative ontology editing and management.

The open source nature of the software is considered an important aspect with the ability to plug new functionalities.

In a Slido poll, an interesting point was raised: if ontology engineering should follow software engineer processes then how do you do ontology testing and quality evaluation as you do in software engineering?

A discussion follows about unit tests for ontologies. One partner does it, but notes that it is tedious. There's a lot of work on that already.

Tools available to do such testing include Themis (http://themis.linkeddata.es/) for testing based on requirements and OOPS! (http://oops.linkeddata.es/) for general checking.

The problem with OOPS that the tests are defined beforehand, and it is not possible to set your own tests. It is suggested that making it open source would solve the problem. OOPS also uses third party systems, so something like a plug-in approach could also work for some cases.

A partner is developing a test-supporting tool (https://w3id.org/OWLunit) which is still an early prototype, but aims at integration into a tool like Protégé or similar.

OntoPortal is raised as a kind of Protégé of ontology repositories (eg: https://iofportal.ncor.buffalo.edu/ontologies). One partner is testing OntoPortal for industrial ontologies. Another is using it to build a portal for cultural heritage ontologies. There is a 'testing instance' (http://arco.istc.cnr.it:8081/) but it is in early stages and very limited. OntoPortal can be used for searching for ontologies as well as publishing them.

Partners suggest a later discussion to evaluate whether it is possible to join forces in order to build an integrated tool for testing, such as a Protégé plugin.

One partner recalls VoxPopuli as a beneficial platform and suggests that perhaps OntoCommons can do something towards retrieving that.

A point is raised about the fact that the cost to create mappings is not the only significant factor, but also to keep mappings up to date without the original author of the pipeline. This is seen to be a crucial point from a longer-term service perspective and flexibility.

One partner presents a metadata schema for ontologies used to evaluate their FAIRness. This raises a discussion about best practices. In https://w3id.org/widoco/bestPractices several suggestions were collected that are recognised in WIDOCO when documenting ontologies.

Discussion about what a company should do if they want to start using and developing ontologies. There are many tools/systems that would be great having in Protégé but the interaction with or

design of Protégé plugins have some constraints. It is not clear whether you can have the interaction as you like or you are restricted to Protégé options…

Guest speaker who was unable to attend provided a link to an earlier similar presentation: https://www.youtube.com/watch?v=QkgAFKL26Vg

A suggestion is made that there are some key functionalities that 'change your OE-life' if they are integrated in the same environment (e.g. requirements-modeling-testing). Others can be external (visualisation, transformation). OE community should share forces and give a new go to Protégé (also changing its core parts can make sense, if evaluated feasible).

A partner wonders if re-thinking the whole thing from scratch would not make more sense - and to inherit some ideas from OntoClean. An alternative view is that going towards continuous integration makes more sense unless improving the system costs more than re-starting from scratch even though it currently entails a great deal of workarounds.

In favour of re-thinking the whole thing is that starting with a high-level architecture model allows for determining which modules and services need to be included, and that architecture can then be evaluated which part can be reused from Protégé. In addition, trying to reuse and repurpose what already exists, which comes from years of evolution, might be harder than rethinking it with the environment and technology we have now. At least, starting from scratch can provide an important way to identify requirements/needs, then consider whether Protégé meets them.

With respect to OntoClean, the crucial novelty is the notion of Relator (a reified relationship)

One solution would be to question the need for a single tool and think instead more towards an "ecosystem of tools", meaning a study of the distinct kind of services which are needed, the provision of a theoretical API for each of those services, and implementation of a global proof of concept with each module remaining independent with its own PI.

But in ontology engineering work you need some of the things to be supported in an integrated environment, or at least your life is certainly easier if you have it. Not everything perhaps, but both “one system for everything” and “one system for each task” are a failure.

Questions raised by this: Is it so hard to improve the plugin system? Is it so hard to develop plugins? Do we want to get rid of Java? Also, how long would it take to re-implement what it's there that we want to keep? Is the Protégé plugin system still a custom layer on top of OSGi?

Although Protégé has many limits, it's also the most used tool, it has supported us well so far and it can be extended. Protégé plugin development is a steep learning curve when starting out.

A proposal for something similar to the EdgeX Foundry founded by the Linux Foundation, which is a big open source community developing software for edge computing. It can be seen as a platform where multiple software solutions are developed in different working groups to build a whole open source stack and to standardize communication tools for various systems.

Some ideas on storing ontologies in git repositories: - when loading, editing, then saving back an ontology document in Protégé, the way the data is presented in the source is often messed up, and the git diff algorithm thinks the whole document has been changed. It would be nice to either: 1. have Protégé respect the original arrangement as much as possible, or 2. propose a new git diff algorithm that works on the level of triples.

The theory of endurant types is an extension of OntoClean (with some small differences). However, here the OntoClean distinctions also apply to reified properties in general (relators, modes, qualities). The theory of relations can be seen as "OntoClean for Relations".

We need a way to experiment with relators against the large repositories of knowledge patterns and frames that are currently used for interoperability, or are discovered: it is not only a top-down problem.

OntoUML as a tool providing support for dynamic modeling (mainly for UFO-A and UFO-B) together. There is a plugin for visual paradigm (https://github.com/OntoUML/ontouml-vp-plugin). They made a lot of updates recently and new release will include stereotypes for events and situations as well as the model cluster feature. The tool is constantly evolving and users are asked to keep their installations up-to-date. Additionally, feedback is not only welcome but very important for the evolution the tool in a way that meets user expectations. It is freely available and open source and can support and integrate other people's research on OntoUML in the future as well. It appears as a free trial and not as open source, but this is because it currently relies on a proprietary tool (Visual Paradigm) as the current SUPPORTED front-end, but the front-end is a mere plugin. All of its code is open source and relies on a server to process OntoUML models. Building plugins to other UML CASE tools is entirely possible but outside of current capabilities. Visual Paradigm has a community version

(https://www.visual-paradigm.com/download/community.jsp) which is free for research/academic use.

Ontology engineering is much like software engineering, but we wouldn't want to put it on the shoulders of those who should be more prone to understand the practice.

It is important to consider which feature can be taken from general or other purposes system and we can reuse for OE. For example, we are using git rather than building a git for ontologies. As ontologists, we should know that configuration management is more than just managing files. So, not only we need ontology configuration management tools but configuration management tools built using proper ontological analysis of the problem

There is a tool allowing annotation of data with ontologies called B2NOTE (b2note.eudat.eu) that is being further developed as a product. It relies on W3C Web Annotation model and allows users to annotate any data that can be uniquely be identified by a URL or other formats

AquaDiva annotates datasets collected within the project, keeping datasets as its original repositories. There is now an extended version that annotates text, tables, images and pdfs.

Solid is based on RDF as underlying technology. Decentralisation in the solid sense requires data to be reusable in several different applications, i.e., it is an interoperability problem. Solid is understood as a technology for ownership of publishing and of data in RDF.

Discussion: What do you most expect from ontology engineering tools?

Tool to validate ontologies through interactions with domain experts in NL:
http://krdbapp.inf.unibz.it:8080/quelo/

Try to open the classical LUBM ontology at http://swat.cse.lehigh.edu/onto/univ-bench.owl Go to the query tab, and push the scramble button once. Try to make sense of the sentence. Re-push the scramble button again, and try to make sense of the sentence again. The tool generates random NL sentences logically consistent with the ontology. If the ontology is crappy, the sentences will be mostly meaningless.

The idea of generating NL sentences that correspond to logical sentences that are provable/satisfiable is a kind of "inverse" competency question

Note that 'crappy things happen' when entity labels are multilingual, or idiosyncratically created (not only when the ontology is ill-designed)

QUELO is a tool finalised in 2011. Documentation at http://www.inf.unibz.it/~franconi/quelo/

Presentation: https://www.slideshare.net/vpresutti/building-the-arco-knowledge-graph-process-experience-and-struggle-with-existing-tools

To validate ontologies we can't just start with intended models, since they depend on the choice of domain and signature. From the practical point of view, a practical evaluation technique is to start from examples and counterexamples in the actual world, then attempt some choices concerning domain and signature, and only at this point consider intended and non-intended models.

The question of the signature is crucial. The question becomes either semantic parsing (how are NL phrases related to the ontology signature) or ontology grounding (how does the ontology signature correspond to the classes and relations in the datasets)?

Quelo works only for English ontologies. There was a huge NL study for that. The signature is in the ontology itself, in the "reverse competency question" approach

[Podcast episode: Identity Graph: The New Customer 360 https://data.world/podcasts/identity-graph-the-new-customer-360/]

Besides psychotherapy, it would be great to present users pictures and movies, just to check if they are going to use a certain term in the same way in front of these examples...

We have a sophisticated method to build a NL lexicon e lexicall semantics from the ontology itself

What about tools for Industrial engineers to use ontologies to solve their problems? If Ontology Engineering is meant in a technical sense then this is an Engineering problem! As for all engineering problems, there is no single tool to solve all problems. It depends on the functional and non-functional requirements at hand.

Developing ontologies for science is a difficult exercise as it must cover a large variety of theories within the same domain and also cope with changes in theory.

To address the issue of reusability, we are working on a set of recommendations to make semantic artefacts FAIR. This work is open for comment: https://zenodo.org/record/4314321#.YFNY17RKhUM

A key missing piece is a search engine on concepts/terms/classes that should rely on existing ontology repositories…

We had early tools such as Swoogle, Watson and Sindice but somehow they all were discontinued. We should try to revive it then! OntoCommons might be the right excuse to do it. Watson was still alive 2 years ago... but those things need active development. We should also leverage other projects such FAIRsFAIR to aggregate ontology repositories. Also patterns, interfaces as well as multimodality for interoperability.

Re: terminology indexing - one partner has a very prototypical codebase which lay dormant over the last year, but would be keen to see resumed. Another has codebase on a system for indexing ontologies from heterogeneous ontology repository that they were planning to use to extend their concept index in B2NOTE for supporting semantic annotation… but funding ran out

This paper was published to describe the initial idea:
https://www.semanticscholar.org/paper/Enhancing-the-Discoverability-and-Interoperability-Goldfarb-Franc/c28e540d201e9418dd59c38d2582683ba55001eb

One of the biggest KGs out there is Wikidata, which is crowdsourced and has a set of own tools for maintainability, validation and curation by their community. They don't seem aware of many of the tools proposed by the Semantic Web community (except perhaps for using custom SHACL shapes). What can we do to reach out to communities like them?

Maybe the problem with the tools is that they are developed in the context of a project and then run out of funding?

That's what was done with Grafo/Capsenta/data.world. Took the learnings from academia/science and productized it

No Magic / Dassault has industry-grade tools. In particular, the Cameo Concept Modeler. There is a single integrated tool for software development because tools must be adherent to the process, methods... which, in turn, depend on functional and non-functional requirements of the software at hand. Different tools support different processes that are appropriate for different software products...

Plurality of tools are not a problem. The problem is the convergence to a single tool that does not support well a large portion of ontologies as products; the fact that it is hard to use different tools together; the fact that there functionalities that are not properly support by any of the tools. And there is no single integrated tool for software development either.

KG is mainly data, but Wikidata has come up with an "ontology" of some sort by themselves.

CYC is proprietary (problem 1) + there are issues with size, coverage, approach. It is time of creating a consortium supported by industry to develop a Ontology tool, which attend the requirements of that industry: simplicity, integration with industrial envivornment, etc.

A software product line for ontologies tools from where we could derive “products” by orchestrating the right tools ...

Academy has distinct goals, consortium of industry + academy are being used for this kind of complex development. This is one lesson that we could leverage on the experience of other communities - it has been more useful than trying to find the fully equipped tool. There is a crucial role of undergraduate education.

Even basic conceptual modeling education often relies on ER diagrams as defined 30-40 years ago (ignoring the results of that community as well). Indeed, modelling is dispensable in computing education (and beyond).

Validation is also a huge problem, and we may not share a common notion of what it means to be a good ontology... perhaps because ontologies are not considered as data. Hence the effort on FAIR Semantics.

A main concern about ontology reuse is not really technological, because you can do it more or less easily with Protégé. The problem is the time spent in understanding someone else's ontology and

evaluating whether it is worthwhile for a project. We should find a way to better communicate ontologies to make it easier for others/us to reuse. It's a people/process problem. Not technology.

Do you see ontology modelling as eventually replacing databases? It depends on what people mean by conceptual modeling...There are many conceptual modeling efforts that are about semantic inteoperabilty When starting to get involved with ontologies it feels like a steep learning curve and a long way to go to be able to build a good one. Hard to be sure which top level ontologies to use or middle level to use. The whole technology stack is pretty complicated and needs a long time to actually get started. With sotware developing or databases it's possible to learn to use a new technology in just a few months.

There are ongoing H2020 projects that are developing ontology-based open frameworks for addressing industrial problems, for data documentation/navigation but also for setting up and execute workflows in e.g. simulation, production, business decisions. If we are restricted to what existed 40 years ago then sure...but the point is that there are techniques developed there that address many of the problems we have to deal with in OE. In fact, in a sense, Ontology Engineering overlaps with Conceptual Modeling (it can be thought as a subtype or as something that includes CM...)

Semantic Web community is quite fragmented and very often people are working in silos... which is ironic as SemWeb aims at removing data silos but we are ending up building semantic silos. Of course validation cannot just be done on a logic basis. It should be done with respect to the ultimate requirement of an ontology, which is to make sure that different users use the same term in a consistent way in front of actual examples and counterexamples.

It is really hard to resolve all the different views on semantics. Shouldn't we try to build a flexible framework allowing to cope with the different approaches?

Paulheim, Heiko, and Aldo Gangemi. "Serving DBpedia with DOLCE–more than just adding a cherry on top." International Semantic Web Conference. Springer, Cham, 2015.
http://www.heikopaulheim.com/docs/iswc2015.pdf

https://ontocommons.eu/news-events/events/top-level-and-mid-level-ontologies-multi-disciplinary-workshop