DLP: a Web-based Facility for Exploration and Basic Modification of Ontologies by Domain Experts

Luca Mazzola
DFKI - German Research Center for Artificial Intelligence
Saarbrücken, Germany
Luca.Mazzola@dfki.de

Patrick Kapahnke
DFKI - German Research Center for Artificial Intelligence
Saarbrücken, Germany
Patrick.Kapahnke@dfki.de

ABSTRACT

One of the main problems for the practical use of tools and technologies of the Semantic Web is the difficulty for a non-expert user of conceiving, analyzing, extending and merging ontologies. Despite the various existing approaches for representing, editing, profiling and comparing ontologies, no integrated solution is available for domain experts. In this paper, we present an initial web-based tool developed to partially s solve this issue, by simplifying the exploration, modification and profile creation of already existing annotated ontologies. DLP offers the functionality identified as fundamental for enabling a domain expert to start working and extending a partially defined semantic data source, lowering the entry barrier for learning the technicalities behind a standard ontology. Additionally, the tool allows ontology modeling experts to interact with the semantic source using the standard SPARQL language.

CCS CONCEPTS

• **Information systems → Web Ontology Language (OWL)**; *RESTful web services*; Browsers; • **Computing methodologies → Ontology engineering**;

KEYWORDS

Ontology editing, Ontology profiling, Ontology exploration, RESTful SPARQL abstraction, Concepts similarity function, Web interface

ACM Reference format:
Luca Mazzola and Patrick Kapahnke. 2017. DLP: a Web-based Facility for Exploration and Basic Modification of Ontologies by Domain Experts. In *Proceedings of 19th International Conference on Information Integration and Web-based Applications & Services, Salzburg, Austria, Dec. 2017 (iiWAS'17)*, 4 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Despite the proliferation of ontologies and tools for manipulating them, usages of Semantic Web (SW) techniques for industrial application is still limited and devoted to some specialized tasks. This is probably also due to the difficulties that non-experts in the semantic domain face to develop, or even only extend a semantic source such as an ontology or a linked data set.

In our experience, as also reported in a work presenting the process and development of a public ontology for some specialized applications into the manufacturing domain [8], the final users tend to assume a passive role, somehow limited to the initial informative talks and the final result approval, being basically absent from the active development part. The problem is twofold: on one side, there is the difficulty of understanding the current content of the ontology, in particular if it is an intermediate product developed by someone else. This is the case most part of applied projects. On the other side, the "exploration" of the semantic data source is typically hard. This last factor is multifaceted: firstly, the serialisation formats of OWL2 were not designed for human readability and secondly a statement can be quite complex and can involve definitions not consecutive inside the textual representation of the ontology, in particular when defining restrictions and annotations.

During the project CREMA[1], a tool called DLP (**D**eveloping onto**L**ogies by **P**rofiling) was developed to overcome these issues with respect to the specif needs. We are going to present DLP basic ideas and features in this document. While the basic concepts of the Semantic Web were clear to most partners involved in CREMA, they were not experts in semantic technologies and in modelling ontologies. Nevertheless, the definition of basic semantic data sources and their customisation to the use cases was a precondition for the objective of an elastic cloud-based processes enactment [9], as service planning and optimisation heavily relies on the semantic annotations of services, tasks and data streams.

The rest of the paper is organised as follows: in section 2 we present the situation with respect to ontology editing and to exploration support tools. Then, section 3 describes our approach and our simplified visualisation for ontology exploration; followed by section 4 reporting on some implementation details and an initial evaluation. Eventually, section 5 briefly closes the work by drawing some initial considerations about DLP.

2 EXISTING SOLUTIONS

A recent survey [11] shows the interest of the ontology engineering practitioners towards the collaborative and community-driven development. Nevertheless, many tools are designed and implemented assuming the participants in this activity are familiar and comfortable with the technicalities of the standards for the Semantic Web (such as *OWL2*). Whether there is a lot of efforts in supporting effectively this type of expert users, with methodologies and tools, we focused on domain experts untrained in semantic technologies.

[1]CREMA is an EU-H2020 RIA project and the acronym stands for "**C**loud-based **R**apid **E**lastic **MA**nufacturing" and its website is http://www.crema-project.eu

iiWAS'17, Dec. 2017, Salzburg, Austria

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

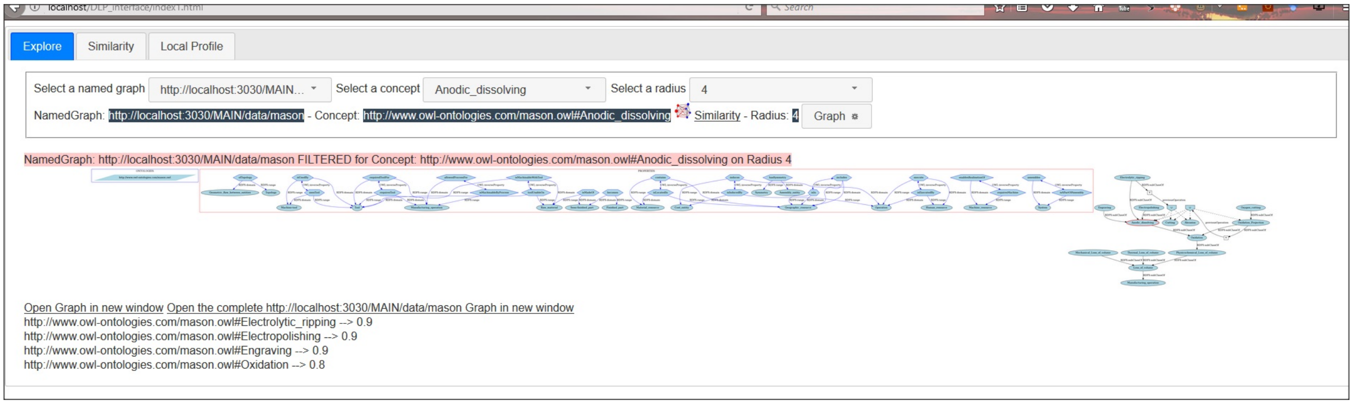

Figure 1: The main interface of DLP. The tabs on the top allow the user to switch between the public ontology (Explore), the similarity exploration, and the local user profile. The commands in this tab allow the selection of a named graph, a concept inside it and a radius, to create a graphical representation of the ontology segment, as partially shown in Fig. 2. To easy the visualisation the created SVG representation can be opened in a new browser tab, and some similar concepts based on our measures are reported at the bottom of the interface.

The objective was to partially bridge the gap towards their capabilities, to basically use ontologies in their knowledge elicitation and organisation tasks.

For the usage/manipulation of semantic data source, the most prominent tool is probably Protégé [10]. OBO-edit [3] is another editor and it supports RESTful-based interactions. However, no concepts of user profiling and metadata management are present in both. Our solution provides a RESTful interface to abstract from SPARQL syntax, for some basic operations, including the creation of user profile and the semi-automatic management of metadata (such as creation and modification time, subjects covered, visibility level).

On top of this, we offer a minimal web-based UI to support the unexperienced user in exploiting the defined RESTful interfaces, enhanced by a basic graphical representation of ontology parts, defined by a user-controllable central concept and an adjustable radius of relationships. This differs from existing approaches that concentrate either on the web UI (such as WebProtégé [12] or Ontofox [13]) or on the ontology representation capability.

For the representation of semantic data sources multiple other tools exist, but all of them aim at the most comprehensive and extensive coverage of the full OWL2 constructs set, while our own lightweight implementation stresses the possibility of control by the user in recursive explorations by recentering and zooming, and the immediateness of the interaction and information presentation.

Well-known tools for ontology representation include VOWL2 [7] and Graphol [2], which concentrate in fully covering the OWL2 constructs. The first one is a visual language and a tool using a force-directed interactive abstraction, while the latter one provides an UML-based graphical notation for the same purpose. Differently, OntoSphere [1] provides an alternative approach for representing concepts and their relationships in terms of a 3D sphere encoding. A more extensive and detailed review of existing approaches to ontology representations can be found in [6].

3 DLP: OUR SOLUTION

Despite the existence of well established solutions for each single presented issue, no tool for providing an integrated environment with the three indicated aspects of SPARQL abstraction, subjects based-profiling and intuitive exploratory representation of a selected part of the ontology is available, to the best of our knowledge. For this reason, we developed a solution that seamlessly integrates these aspects.

Nevertheless, DLP does not implement a fully cooperative distributed management for conflict resolution, as we preferred a more lightweight approach. It is based on the reduction of conflict possibilities by local editing of the subpart of the publicly available ontology. Despite not solving the possibility for "dirty writes", this approach concentrates the potential conflicting operation to the final merge operation, instead of spreading it on every single modification. This solution was sufficient for the project, where a restricted well coordinated group of domain experts was involved.

As a starting point for developing our DLP facility we identified a set of abstract interfaces, to encapsulate the most common SPARQL operations. Secondly, we designed an AJAX web-based simplified minimal graphical user interface (GUI) for ontology editing, adding the possibility to use profiling and subsetting of existing ones and with the possibilities to merge back the produced result.

3.1 RESTful API for ontology editing

Following a common choice for providing interoperability and reusability, we decided to adopt a RESTful based technique in implementing the ontology editing interfaces.

As can be noted, the included functions abstract from the basic SPARQL update functionality, grouping them to produce more user-oriented functions, that our non-expert indicated as preferred to have when interacting with a SPARQL endpoint. This list includes sparse CRUD (*Create*, *Read*, *Update*, *Delete*) operations at the level of an ontology, of the user profile, of its metadata, plus operations to provide search functionality over concepts and instances. Despite not being complete, the list represents the basic operations our users requested and used when dealing with the domain ontologies of the CREMA project, called CDM-Core [8].

One interesting point is connected with the similarity search functions, as they allow a user to search for the ordered list of top-k similar entries in the ontology or to compute the similarity amongst two given concepts.

3.1.1 Similarity computation. For similarity computation, a simplified approach was proposed and agreed on with the user partners of the project. Each function associated with the "Similarity computation" label implements a simple ontological structure-based similarity measure on the directed CDM-Core graph, without logic-based reasoning such as logical unfolding of concept definitions in the ontology and information-theoretic measurements, and is restricted to exact name matching. Based on a modular approach, other approaches can be implemented by simply changing the relevant code class. For state of the art information on similarity operators in general, and semantic similarity measures in particular, we refer to [4] and [5], respectively.

In practical terms, the similarity measure is based on structural information, weighted by their meaning, but limited to a subset of the OWL2 construct and to a certain user-influenceable distance from the analysed concept. The type of directed relations of paths in the CDM-Core from the given concept (class) to other concepts (classes) or instances in the ontology are weighted. The single weights are determined based on some experiments in our semantic data source, but are not guaranteed to generalise well, as they clearly depend on the interconnectivity level and the richness of the particular data source. Nevertheless, we will give an intuitive explanation of the chosen weights to help readers tuning it based on particular needs. It is restricted to the relations owl:equivalentClass for asserted equivalent concepts, rdfs:subClassOf for its parent and children concepts, rdfs:type for types of its instances, and owl:restriction statements for related complex concepts. The weights of these relations are set by default within the interval [0,1].

The similarity computation with the default weights prefers, in general, concepts that are equivalent [1.0] to or more specific (children) [0.9] than the given concept over those which are more generic (parents) [0.8]. Besides, the similarity of a concept in the path from the given concept decreases with the path length. Relevant concepts are preferred over relevant instances [0.7] and complex concepts connected to the given concept in the concept taxonomy (class hierarchy) [0.5] of the given named graph in the ontology. Eventually, the semantic similarity value in [0,1] of a given concept to some concept or instance in a directed relation path is computed as the sum of multiplied weights of the relations. In the second tab (called "Similarity", as shown in Fig. 3) the use has the possibility to query the semantic source for concepts and instances similar to the one given, into a selected named graph. In this context, the domain expert has also the control over the weights used for each one of the five categories considered.

3.1.2 Ontology segment representation. The approach is focused on giving an intuitive first insight, and is based on a self-written PHP class that interprets a segment of RDF/XML ontology representation to create an abstract view.

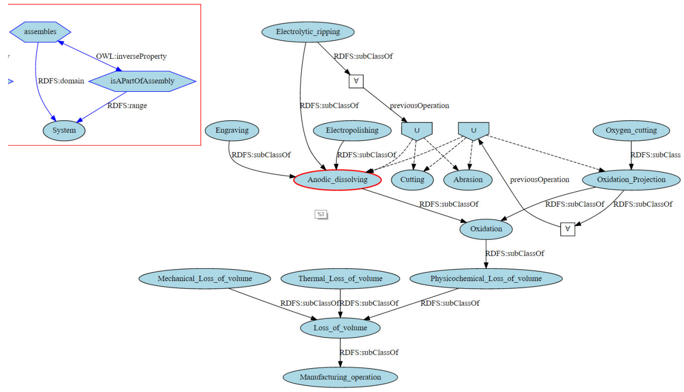

Figure 2: Example of our simplified visualisation of the context of the "Anodic dissolving", from MASON. Partially visible in the left box is the relevant property tree relevant.

This abstraction is then translated on the fly into a SVG file. The coverage of the full set of OWL2 construct is indeed very limited, because the objective is giving a first and quick interpretable figure for a domain expert. The produced graphical representation can then support the iterative exploration of the semantic concept context limited to a selectable radius, without relying on any semantic format expertise. An example for the simplified graphical representation is shown in Fig. 2.

3.1.3 Metadata Profile. Another pecularity of the DLP tool is the managment and semi-automatic update of the metadata set associated with the ontologies. It is based on the DublinCore metadata initiative vocabulary[2], and allows the user to indicate information such as the covered subject, the visibility level, the creation, modification and an optional release date, which original ontologies were imported in the initial profile creation, etc. Its added value is the possibility to merge back the modification in a reasoned way, affecting only the relevant part of the publicly released semantic data source, and to maintain track of the contributor and the provided contribution. Furthermore, it allows a user to discover (and take corrective actions) if there are updates for his/her profile (or even of a part of it). Nevertheless, if a user modified its local profile without committing it back, DLP does not automatically support a merge of the local contribution into the new ontology version. This functionality can be accessed by a user selecting the ''Local Profile", that allows to create (if not existing) a profile, update it, explore its parts (metadata and actual semantic content), merge it back to the public release, or delete it. This provide the working space for the domain experts, to test modifications in a isolated environment.

4 IMPLEMENTATION DETAILS

This section will give a very high level overview of the DLP component. It is implemented as a web application using (x)HTML and PHP, a scripting language which is interpreted inside the Apache HTTPD web server. All the application logic resides in the scripts, where the access logic is implemented with a set of rewrite rules inside the VirtualHost settings of the web server. The client side

[2] http://dublincore.org/documents/dcmi-terms/

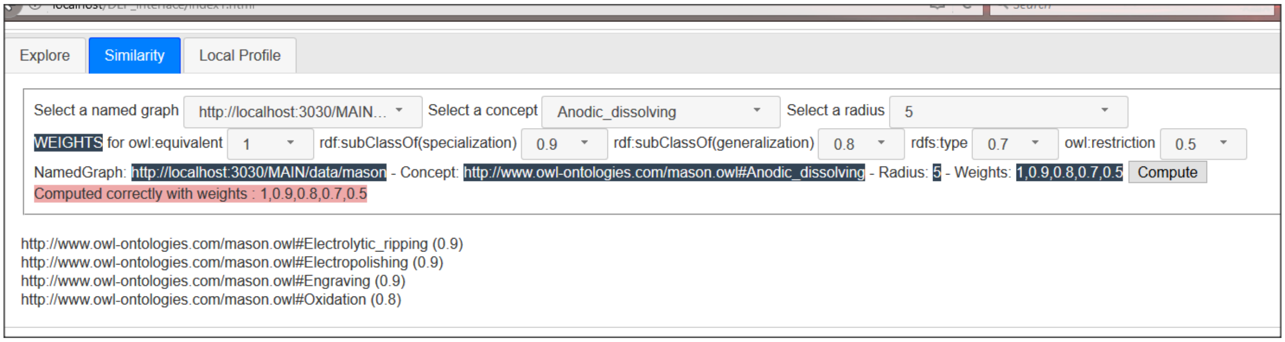

Figure 3: This tab allows the user to retrieve and rank concepts and instances similar to a given one, using our simplified similarity function. After selecting the named graph, the concept and the radius, a user can give personalized weights to the five different semantic properties we consider and launch the computation. After the reuslts are ready, they will be showed i nthe bottom part of the interface, together with their [0,1] similarity measure, in brackets. Clicking on any concept in the similar list will bring back the interface into the "Explore" tab, where a new iterative step can start for further explorations.

interactions are based on an AJAX approach. As storage layer, the data of the public ontology and user-specific profiles are stored in an Apache Jena RDF triple store. Internally, the Jena Fuseki2 module is used for RESTful interaction with the SPARQL endpoint of the triple store. All the software used for the implementation of the DLP component has open source licenses.

The developed tool is a demonstrator, both in the capabilities and in the development approach, as generally happens in research and innovation projects. The solution is published as a Docker (see https://www.docker.com/) self-contained image, under the name *SW-DLP* as AGPLv3 public code at **https://sw-dlp.sourceforge.io/**

4.1 Initial evaluation

Together with the domain experts we run a first evaluation step, asking the user to rank on a likert 5 points scale the usefulness and intuitiveness of the DLP tool. The aggregated results seem to suggest a positively polarised impression (respectively an average of 4.0 and 3.80) but also reported some critical points that can and should be improved for a real usage of the tool by domain experts untrained for semantic technologies and standards.

5 CONCLUSIONS

We developed and presented a demonstrator tool to partially solve the issue of semantic data source editing by non-experts. The added value of our simplified approach is the possibility of working on a selected subpart of the source relying on a user copy (profile). This approach is particularly interesting for the evolution, adaptation or extension of a partially defined ontology to fit a specific use case, as it is frequently the case for applied projects. On top of the editing, we also support a simple but still informative and immediate graphical representation of a part of the semantic source and the computation of an initial similarity function with user-selected weights. Moreover, we support the iterative exploration of a user-defined context radius for a concept, reducing the cognitive overload for domain experts. From the management point of view, we adopted a simplified approach based on metadata for minimal collaborative distributed ontology evolution. The DLP tool was effectively tested with the CREMA domain experts and the user partners and successfully supported the definition of the core ontology for CREMA.

ACKNOWLEDGEMENTS

This work was partially financed by the European Commission within the CREMA H2020-RIA project, under the agreement 637066.

REFERENCES

[1] Alessio Bosca, Dario Bonino, and Paolo Pellegrino. 2005. OntoSphere: more than a 3D ontology visualization tool.. In *Swap*. Citeseer.

[2] Marco Console, Domenico Lembo, Valerio Santarelli, and Domenico Fabio Savo. 2014. Graphol: Ontology Representation through Diagrams.. In *Description Logics*. 483–495.

[3] John Day-Richter, Midori A Harris, Melissa Haendel, Suzanna Lewis, Gene Ontology OBO-Edit Working Group, et al. 2007. OBO-EditâĂŤan ontology editor for biologists. *Bioinformatics* 23, 16 (2007), 2198–2200.

[4] P Deepak and Prasad M Deshpande. 2015. *Operators for Similarity Search: Semantics, Techniques and Usage Scenarios*. Springer.

[5] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. 2013. Semantic measures for the comparison of units of language, concepts or instances from text and knowledge base analysis. *arXiv preprint arXiv:1310.1285* (2013).

[6] Maurizio Lenzerini, Diego Milano, and Antonella Poggi. 2004. Ontology representation & reasoning. *Universitá di Roma La Sapienza, Roma, Italy, Tech. Rep. NoE InterOp (IST-508011)* (2004).

[7] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. 2014. VOWL 2: user-oriented visualization of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 266–281.

[8] Luca Mazzola, Patrick Kapahnke, Marko Vujic, and Matthias Klusch. 2016. CDM-Core: A Manufacturing Domain Ontology in OWL2 for Production and Maintenance. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016)*. SCITEPRESS - Science and Technology Publications, Lda, Portugal, 136–143. https://doi.org/10.5220/0006056301360143

[9] Luca Mazzola, Patrick Kapahnke, Philipp Waibel, Christoph Hochreiner, and Matthias Klusch. 2017. FCE4BPMN: On-demand QoS-based Optimised Process Model Execution in the Cloud. In *Proceedings of the 23rd ICE/IEEE ITMC conference*. IEEE.

[10] Natalya F Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W Fergerson, and Mark A Musen. 2001. Creating semantic web contents with protege-2000. *IEEE intelligent systems* 16, 2 (2001), 60–71.

[11] Elena Paslaru Bontas Simperl and Markus Luczak-Roesch. 2014. Collaborative ontology engineering: a survey. *Knowledge Eng. Review* 29 (2014), 101–131.

[12] Tania Tudorache, Jennifer Vendetti, and Natalya Fridman Noy. 2008. Web-Protege: A Lightweight OWL Ontology Editor for the Web.. In *OWLED*, Vol. 432. Citeseer.

[13] Zuoshuang Xiang, Mélanie Courtot, Ryan R Brinkman, Alan Ruttenberg, and Yongqun He. 2010. OntoFox: web-based support for ontology reuse. *BMC research notes* 3, 1 (2010), 175.