Augmenting the Ontology Visualization Tool Recommender: Input Pre-Filling and Integration with the OOSP Ontological Benchmark Builder

Ondřej Zamazal, Marek Dudáš and Vojtěch Svátek
Department of Information and Knowledge Engineering
University of Economics, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic
{ondrej.zamazal|marek.dudas|svatek}@vse.cz

ABSTRACT

Ontology visualization is an important functionality for ontology users. Since many visualization methods have been proposed and implemented, it is not an easy task for an ontology user to select a proper ontology visualization tool based on the user's requirements. In order to recommend an ontology visualization tool, some requirements should be provided manually by the user. However, other requirements could be assessed automatically. This demo paper presents a partial automation of the pre-existing Ontology Visualization Tools Recommender input; the recommender knowledge base is also updated with three new ontology visualization tools. Further, we present an integration of the Ontology Visualization Tools Recommender into another web-based tool, Online Ontology Set Picker, where proper ontology visualization can assist in ontology benchmark construction.

1. INTRODUCTION AND MOTIVATION

In order to thoroughly understand the ontology content of an ontology, the users usually need adequate visualization functionality. Since many visualization methods have been proposed and implemented, it is not an easy task for an ontology user to select a proper ontology visualization tool based on his/her requirements.

In 2014 we implemented the "Ontology Visualization Tools Recommender" (OVTR)[1] [3], in the form of the knowledge base for the NEST expert system shell [1] plus specifically tailored web page. In the current demo paper, in response to the usage of OVTR in the meantime, we introduce several improvements aiming at better user experience.

First, in order to recommend an ontology visualization tool by OVTR, all the requirements have to be provided by the user as input for the consultation with the knowledge base. However, some requirements could be collected automatically, which would make the visualization tool recommendation faster and less tedious for the users. Second, if the user wants to visualize more ontologies for the same purpose, s/he currently has to perform multiple consultations in OVTR. In this case aggregated recommendation for a whole bunch of ontologies is beneficial. Third, we assume that in practice it is often useful to properly visualize ontologies during their usage in different ontology tool workflows. In this respect we couple OVTR with a particular workflow, that of constructing an ontological benchmark for tool testing by means of a tool called Online Ontology Set Picker (OOSP).

This demo paper covers those three issues by a partial automation of the OVTR input and by an integration of OVTR into OOSP, where proper ontology visualization can help with ontology benchmark construction.

While the OVTR core is presented in Section 2, Section 3 explains how we extend OVTR in terms of its input automation along with a description of two envisaged scenarios. Section 4 presents the OOSP tool along with a scenario describing the integration of OVTR into OOSP. Section 5 mentions some related work, and Section 6 wraps up the paper with conclusions and future work.

[1]http://owl.vse.cz:8080/OVTR/

SEMANTICS 2016: Posters and Demos Track September 13-14, 2016, Leipzig, Germany

2. OVTR OVERVIEW

While the frontend of OVTR is implemented as a Java web application, the backend is supported by the NEST web service working with knowledge captured in the NEST knowledge base. The NEST expert system shell covers the functionality of traditional compositional rule-based expert systems with uncertainty handling in a standard range between -1 (certain FALSE) and +1 (certain TRUE)[2] where 0 means irrelevant and does not have any effect on reasoning. NEST employs a combination of backward and forward chaining and it processes uncertainty according to the algebraic theory of Hajek [5]. For capturing task-specific knowledge for reasoning about ontology visualization, the knowledge base contains 8 attributes (2 binary, 5 nominal set and 1 numeric), 36 propositions and 32 compositional rules.[3] Altogether there are 14 *visualization tools* for recommendation, represented as output propositions of the knowledge base: Protégé Entity Browser, TGVizTab, Ontoviz, Jambalaya, OWLViz, Ontograf, OwlGrEd, Ontology Visualizer, KC-Viz, SOVA and TopBraid. Further, there are two attributes aggregating the relevant answers from the user: *Use case category* (editing, inspection, learning and sharing) and *OWL* (expressing the importance of particular OWL features). Finally, there are five attributes representing the possible user's answers to various questions:

[2]The range of the interval can be extended by multiplication of both sides. The knowledge base for OVTR uses the interval from -3 to 3.
[3]All details are presented in [3].

- *Complex classes* reflects the importance of anonymous classes based on various OWL constructs such as union, complement, intersection etc.

- *Intended usage* includes eight predefined use cases [3] such as making screenshots, checking domain inconsistencies, checking structural inconsistencies, reusing an ontology, adapting an existing ontology, developing a new ontology, searching for the right ontology and analyzing mappings of the new ontology.

- *Ontology size* contains three fuzzy intervals: for 'small', 'medium' and 'large' ontologies.

- *OWL features* reflects the importance of particular OWL features (object properties, inter-class relationships, datatype properties and property characteristics), see Table 1, aiming to infer the overall importance of OWL support in the visualization.

- *Favorite ontology editor* captures the user's preference for some (freely available) ontology editor: Protégé 3, Protégé 4 or Neon Toolkit.

As a collateral contribution to this demo paper we extended the original OVTR knowledge base presented by Dudáš et al. in [3] with three more tools: Graffoo[4], WebVOWL[5] and Ontodia.[6]

3. INPUT AUTOMATION FOR OVTR

The user of OVTR should specify information about the above mentioned attributes in order to get a recommendation.[7] Beside others, the user should also assess the importance of particular OWL features and specify the ontology size. Since the user probably has some particular ontology in his/her mind, our extension to OVTR aims at supporting the user by automatic assessment of importance of OWL features involved in the knowledge base (complex classes, object properties, interclass relationships, datatype properties, and property characteristics) and ontology size.

While obtaining the ontology size is straightforward,[8] in order to assess the importance of OWL features we must first interpret the lingustic notion of importance. For instance, let us consider the question about complex classes in OVTR: *Are complex classes important for the ontology visualization?* The user is expected to provide the answer as a weight in interval between -3 and 3.[9] There are at least two possible interpretations of the notion of *importance* in this case. We could either view this in terms of *absolute numbers*, i.e. the higher number of complex classes the ontology has, the more important they are for the ontology, or in terms of *relative numbers*. In the case of relative numbers it would have to be specified to what ontology feature it is related, e.g., we can relate complex classes to the size of the ontology (as mentioned above). In both cases we can rescale the (absolute or relative) numbers to the range between -1 and 1 using the following two equations:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad | \quad x' \in [0, 1] \tag{1}$$

$$x'' = (x' - 0,5) \times 2 \quad | \quad x'' \in [-1, 1] \tag{2}$$

Equation (1) refers to the min-max normalization as described, e.g., by Han et al. [6] where x is a value of the ontology metric of given ontology, $min(x)$ ($max(x)$) refers to minimum (maximum resp.) of the ontology metric within selected ontology pool and x' is higher or equal to zero and lower or equal to 1. In order to scale into interval [-1,1] we use Equation (2). Maximum and minimum components in Equation (1) can be computed with regard to some ontology collection for the respective OWL features. This kind of automation is involved in three scenarios.[10]

Scenario 1.

Recommendation of an ontology visualization tool based on a given ontology aims at accelerating and easing the process of recommendation by the partial automation of its input, otherwise not deviating from the original OVTR usage scenario. First, the user can upload an online ontology and select the ontology pool context according to which min-max normalization will be performed. Second, the user can inspect the size of the ontology and the absolute numbers of related OWL features. Third, importance weights are computed for each OWL feature. Next, the user is expected to input further answers if applicable. The user can also revise the computed importance of OWL features.

In our preliminary implementation of importance weight computation we use absolute numbers of related OWL features, and we use the maximum and minimum values (for Equation (1)) according to the ontology pool selected by the user.

A screencast demoing the use of OVTR including extensions introduced in this paper is available from the OVTR website.[11]

Scenario 2.

Recommendation of an ontology visualization tool based on a given collection of ontologies aims at supporting the situation when the user needs to visualize *multiple* ontologies. The ontologies may differ in their metrics, but other ontology visualization requirements can be shared (e.g., the favorite ontology editor or intended usage). It is a variation of Scenario 1 in the sense that instead of one ontology a collection of ontologies is to be uploaded on input. The consultation is performed for each ontology in the collection and then the recommendation for each ontology as well as an aggregated recommendation is provided to the user.

While Scenario 1 is fully implemented, support for Scenario 2 is only in progress for the moment, because of performance issues potentially incurred by a possibly larger number of aggregate consultations.

[4]http://www.essepuntato.it/graffoo/
[5]http://vowl.visualdataweb.org/webvowl.html
[6]http://dev.ontodia.org/
[7]A traditional strong point of expert systems is that it is not necessary to provide all answers in order to receive a reasonable recommendation.
[8]The knowledge base in OVTR considers size as $|Properties \cup Classes|$.
[9]When submitted to NEST they are converted to the <-1, 1> interval.
[10]The first two immediately follow, while the last one is presented in Section 4.
[11]http://owl.vse.cz:8080/OVTR/

Table 1: Components of OWL features.

OWL feature	relevant OWL components
complex classes	$Enumeration \cup Intersection \cup Union \cup Complement \cup Cardinality \cup Existential$ $quantification \cup UniversalQuantification \cup ValueRestriction \cup SelfRestriction$
interclass relationship	$Disjoint \cup Equivalent \cup Subclassof$
object properties	$ObjectProperties$
datatype properties	$DatatypeProperties$
property characteristics	$Transitive \cup Symmetric \cup Functional \cup Inverse$

4. OOSP ENHANCEMENT BY OVTR

OOSP is a tool allowing to select, from major repositories, a set of ontologies satisfying a user-defined sets of metrics [11]. It aims at allowing the ontological tool designers to rapidly build custom benchmarks on which they could test different features of their tools. The web front-end allows to specify a broad range of metrics and delivers benchmarks along with their statistics of metrics, including a graph view. Currently it includes the following snapshots from different repositories:

- the *BioPortal*[12] February 2015 snapshot contains 317 ontologies,
- the *Linked Open Vocabulary* (LOV)[13] February 2015 snaphot contains 461 ontologies,
- the LOV January 2016 snapshot contains 509 ontologies,
- the *NanJing Vocabulary Repository*[14] (NJVR) January 2016 snapshot contains 1403 ontologies,
- the NJVR merged[15] January 2016 snapshot contains 225 ontologies,
- and the OntoFarm[16] Jan. 2016 snapshot contains 16 ontologies.

Scenario 3.

Integration of recommendation of an ontology visualization tool based on a given ontology into OOSP. This scenario aims at facilitating the process of ontology benchmark construction in OOSP. Ontologies included in the ontology set selected according to ontology metrics can be subsequently inspected using a suitable visualization tool. The user can use OVTR directly from OOSP, where it is possible to ask for a recommendation of visualization tool by clicking on the "vis hint" link next to each ontology storage code in the selected ontology set. In this case the process of recommendation is fully automatic. The setting of the remaining answers for the consultation is based on several assumptions.

First, it is assumed that the user wants to visualize the ontology in order to "learn the ontology itself", therefore two interrelated *use cases* are selected:

- analyzing an ontology in order to annotate data with (or create instances of) its entities, and
- deciding about the ontology suitability for a specific use case.[17]

Second, the recommended visualization tools should be rather up to date, as well as either easy-to-install or web-based. Protégé 4 (with weight set to 3) is therefore selected as most preferred, followed by Protégé 3 (with weight equals to 2), in case the consultation leads to a toolbox plugin variant.

After receiving the recommendation the user can revise her/his requirements in the OVTR tool and possibly rerun the consultation. The implementation uses the same setting of Equation (1) as Scenarios 1 and 2, except that the maximum and minimum values are currently set according to ontology pool selected by the user during the ontology benchmark construction in OOSP. Since the characteristics of ontology pools differ a lot, the selected ontology pool has large impact on the resulting weights. For instance, in the case of a large ontology from BioPortal, the *bridgmodel*,[18] having 1310 object properties, 2487 compound class expressions, 306 inter-class relationships and 333 property characteristics, the resulting weights for each OWL feature are as follows:

- for the BioPortal context: complex classes=-2.9, inter-class relationships=-3, object properties=3, datatype properties=-3, property characteristics=-1.35,
- for the LOV context: complex classes=2.34, interclass relationships=-2.7, object properties=0.12, datatype properties=-3, property characteristics=3,
- for the OntoFarm context: complex classes=3, inter-class relationships=0.54, object properties=3, datatype properties=-3, property characteristics=3.

Basing the weight computation on the pool used in OOSP has the advantage that the resulting weight is likely to be intuitive for the user familiar with this pool. For example, a user familiar with BioPortal ontologies would not be surprised that *bridgmodel* has not so many complex classes (about 2 thousand) compared to other BioPortal ontologies. On the other hand, with respect to LOV and especially OntoFarm, which mostly contain much smaller ontologies, the overall number of complex classes would be quite high. Since the OVTR knowledge base has been manually designed with respect to the common sense by which visualizing even a few dozens of complex classes is a challenge, relying on some 'cross-pool' generalization might thus be more meaningful in terms of recommendation quality. This is a subject of further research.

[12]http://bioportal.bioontology.org/
[13]http://lov.okfn.org/dataset/lov/
[14]http://ws.nju.edu.cn/njvr/
[15]They are experimental ontologies which were created as a merge of their definitions spreading over RDF files.
[16]http://owl.vse.cz:8080/ontofarm/
[17]These use cases were originally described in [3] and correspond to uc7 and uc8.
[18]http://www.bridgmodel.org/owl/3.2

A screencast demoing the use of OOSP including the OVTR invocation is available from the OOSP website.[19]

5. RELATED WORK

To the best of our knowledge there is no directly related work in terms of automatically recommending ontology visualization tools. However, there is related work regarding recommendation in the visualization domain in general. For example, Voigt et al. [10] aimed at knowledge-assisted visualization recommendation for non-experts who need visualization of a large data set. In this case the visualization knowledge is captured using the VISO ontology. Recommendation is based on mapping the VISO ontology to data sets and visualization component descriptors. The authors further invented their own recommendation (ranking) algorithm. On the contrary, we use the traditional expert system technology for recommendation, and do not focus on visualizing data sets.

As regards surveys of ontology visualization tools, there is a recent brief survey by Bikakis and Sellis [2].

In terms of comparison, our recommender rates the tools by their feature support and our expert insight gained by testing the tools with various ontologies. A different approach is that of comparing the tools according to user-based evaluation. Such an evaluation have been done by Katifori et al. [7], Motta et al. [9], Fu et al. [4] and Lohmann et al. [8]. Each evaluation project selected several (up to four) tools and compared how well and how fast the users could perform specific tasks with them.

6. CONCLUSIONS AND FUTURE WORK

The demo paper presented partial input automation of the pre-existing OVTR tool (with its knowledge base extended with three new tools), potentially leading to its faster usage. The automation is based on the OWL features present in the ontology given on input. The numbers of OWL features are converted to weights expressing the importance of these features. We presented two scenarios where an ontology or an ontology collection is provided as input by the user. The third scenario aims at facilitating ontology benchmark construction in the OOSP tool, by means of recommending the means of visual inspection of a chosen ontology.

In our implementation we used min-max rescaling when converting metrics to weights, however, since it is sensitive to extremes (outliers), we plan to investigate whether some other rescaling approaches might be better. For example, one option could be to use a standard score, *z-score*, as described by Han et al. [6], which indicates whether a value is above or below the mean and quantifies how much. This can also be converted to a probability determining the area underneath the distribution curve below the z-score. The shorcoming of the z-score is that it applies to the normal distribution, whereas ontology metrics usually do not exhibit this distribution.

We plan to implement the second scenario, dealing with an ontology collection on input, in the near future. We also plan to experiment with an alternative computation of importance of OWL features using relative numbers instead of absolute numbers. Further, we plan to keep our knowledge base up to date with regard to new ontology visualization tools.

[19] http://owl.vse.cz:8080/OOSP/

7. ACKNOWLEDGEMENT

Ondřej Zamazal has been supported by the CSF grant no. 14-14076P, "COSOL – Categorization of Ontologies in Support of Ontology Life Cycle". The research is supported by UEP IGA F4/28/2016.

8. REFERENCES

[1] Berka P.: NEST: A Compositional Approach to Rule-Based and Case-Based Reasoning. In: *Advances in Artificial Intelligence*, 2011, 15 pages. Online: http://www.hindawi.com/journals/aai/2011/374250/.

[2] Bikakis, N., Sellis, T.: Exploration and visualization in the Web of Big Linked Data: A survey of the state of the art. 2016. arXiv preprint arXiv:1601.08059.

[3] Dudáš, M., Zamazal, O. and Svátek V.: Roadmapping and Navigating in the Ontology Visualization Landscape. In: Knowledge Engineering and Knowledge Management (EKAW 2014). Springer.

[4] Fu, B., Noy, N. F., Storey, M. A. Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. In The Semantic Web. ISWC 2013 (pp. 117-134). Springer Berlin Heidelberg. 2013.

[5] Hajek P.: Combining functions for certainty degrees in consulting systems. In: *International Journal of Man-Machine Studies*. 22(1), pp. 59-76, 1985.

[6] Han J., Pei J., Kamber M. Data mining: concepts and techniques. Elsevier. 2011.

[7] Katifori, A., Torou, E., Halatsis, C., Lepouras, G., Vassilakis, C.: A comparative study of four ontology visualization techniques in protege: Experiment setup and preliminary results. In: Information Visualization, 2006. IV 2006. Tenth International Conference on (pp. 417-423). IEEE.

[8] Lohmann, S., Negru, S., Haag, F., Ertl, T.: VOWL 2: user-oriented visualization of ontologies. In Knowledge Engineering and Knowledge Management (pp. 266-281). Springer International Publishing. 2014.

[9] Motta, E., Mulholland, P., Peroni, S., d'Aquin, M., Gomez-Perez, J. M., Mendez, V., Zablith, F. A novel approach to visualizing and navigating ontologies. In The Semantic Web. ISWC 2011 (pp. 470-486). Springer Berlin Heidelberg. 2011.

[10] Voigt M., Pietschmann S., Grammel L., Meissner K. Context-aware recommendation of visualization components. In: The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW). 2012.

[11] Zamazal O., Svátek V.: OOSP: Ontological Benchmarks Made on the Fly. In: Workshop SumPre'15 at ESWC 2015.