When Ontologies met Knowledge Graphs: Tale of a Methodology

Romana Pernisch[1,2], María Poveda-Villalón[3], Diego Conde-Herreros[3], David Chaves-Fraga[4], and Lise Stork[1]

[1] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
{r.pernisch, l.stork}@vu.nl
[2] Discovery Lab, Elsevier, Amsterdam, The Netherlands
[3] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
{diego.conde.herreros,m.poveda}@upm.es
[4] Grupo de Sistemas Intelixentes, Universidade de Santiago de Compostela, Spain
david.chaves@usc.es

Abstract. The current state of the art of knowledge engineering lacks proper methodologies to deal with the ever-changing nature of knowledge. In this short paper, we present a first step towards including the changing nature of knowledge in the knowledge graph lifecycle. We extend the LOT ontology engineering methodology to include activities associated with knowledge graph construction, better reflecting how they are engineered in the real world. Further, we analyse how these lifecycles compare to ontology evolution frameworks and what work is there to be done in the future to step from engineering towards full knowledge graph evolution.

Keywords: KG construction, KG lifecycle, methodology

1 Introduction

The constantly evolving nature of knowledge has become a major problem in the way we engineer and publish data on the Web in the form of knowledge graphs (KGs)[5] [5]. We lack methodologies that accurately capture the problem of evolving KGs and at the same time propose how to deal with changing KGs over time. This problem is aggravated by the fact that today we have methodologies for the engineering of ontologies [6,4,2], and KGs [8,1] separately.

Although Ontology 101 [4] specifically includes the activity "creation of individuals", it does not consider today's technologies such as RML, SHACL, ShEx, which are involved in the engineering of KGs. Even LOT [6], one of the newest ontology engineering methodologies, focuses only on the engineering of the schema, or what we refer to as ontology, and does not consider the population with large amounts of data. Works such as those by Radulovic et al. [7], Chaves-Fraga et al. [1] and Simsek et al. [8] have abstracted the process to different levels and varying focus. Radulovic et al. [7] provide guidelines on what steps to take, and

[5] We consider a KG to consist of a Tbox (terminology, schema) and Abox (assertions).

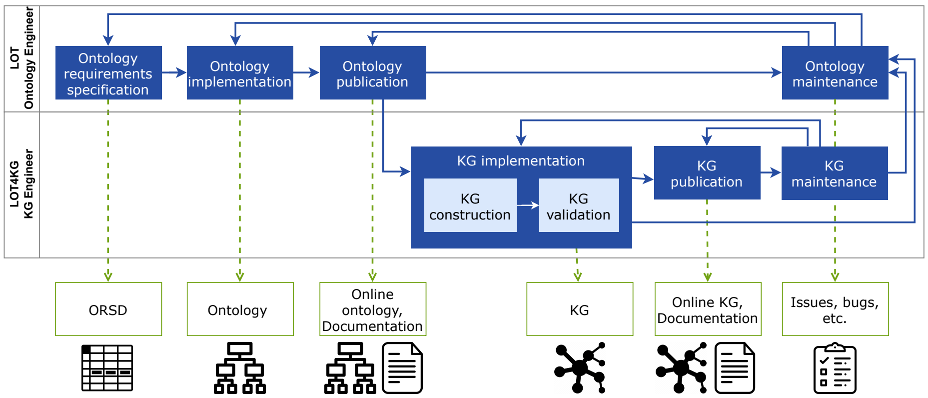

Fig. 1. High-level methodology overview with the LOT methodology [6] (top), and knowledge graph (KG) lifecycle (bottom).

Chaves-Fraga et al. [1] describe the process they used when engineering a KG for research-performing organisations. Simsek et al. [8] abstract the process, which is where we take our inspiration for the proposed methodology.

Therefore, in this short work, we propose to extend LOT into a new methodology, which integrates the engineering of the schema as captured by LOT [6] and adds the engineering of KG into a joint methodology. Then, we discuss our proposal in the context of knowledge evolution and propose future research on integrating it into LOT4KG.

2 Proposed Methodology

The original LOT methodology [6] details the process of *ontology requirements specification*, *ontology implementation*, *ontology publication* and *ontology maintenance*, shown in the upper lane of Figure 1. Our extensions, the KG lifecycle (bottom lane), is described in detail further below.

We identify three high-level activities: *KG implementation*, *KG publication*, and *KG maintenance*, mirroring the LOT ontology lifecycle. The KG lifecycle starts after the publication of the ontology, so there is an activity flow from *ontology publication* to *KG implementation*. Unlike the ontology engineering process, no requirement specification activity is required: the ontology imposes requirements on the KG. *KG implementation* is analogous to the ontology implementation activity and describes the steps taken to construct the KG. We also distinguish lower-level activities: KG construction and KG validation, similar to [8]. During KG construction, we generate relationships between heterogeneous data sources and ontology terms using mapping languages (e.g., RML or SPARQL-Anything). In a separate step, SHACL shapes are generated, which impose constraints on the shape of the KG and are thus used for KG validation. The output of the validation activity may generate a refined version of the KG. These two activities may be divided into more fine-grained activities such as the generation of mapping rules, the transformation of input sources into

RDF and debugging. *KG publication* includes the publication of the KG and its corresponding documentation in human-readable format. The publication also includes not only documentation of the actual KG but also of the associated assets (e.g. documentation of RML mappings or SHACL) Lastly, *KG maintenance* is analogous to that of ontology maintenance. Issues and bugs are collected during a certain period of time, which, in turn, can trigger the implementation and publication of KG.

3 Context of Knowledge Evolution

In ontology and, consequently, in KG evolution, the need for change can come from different sources [11] other than from the process of fixing issues and bugs. These needs for change can be divided into two categories: (i) changes in business requirements, therefore, changes to ontology requirements, and (ii) changes to the underlying application domain, which needs to be represented by the ontology/KG [9]. Further discussion and deliberation are needed, as changes can also come from input sources, affecting the KG construction, depending on the changes and possibly the ontology. Such changes and update activities are not captured with LOT and its presented extension at this point, although the methodologies are circular. Once integrate, we propose to refer to it as LOT4KG. Additionally, KG maintenance faces more challenges in comparison to ontology maintenance because of the technologies involved in KG construction and their less mature tool support. With ontology editors, changing an ontology has been relatively easy for many years, but we do not edit a large KG which was created using RML but rather recreate it. Further, the SHACL validation also requires an update and depending on the changes this can become a costly endeavour.

Therefore, as a research community, we need to evaluate how KG maintenance is done today and how it compares with known ontology evolution frameworks [11,10]. The activities that need to be discussed are distinct from ontology and KG implementation activities to the extent that the engineers are updating the already existing artefact rather than creating a new one. Hence, at the ontology level, we should be able to produce a list of changes [11,3] according to which the KG can be updated. In a KG update, not all mapping rules and validation shapes need to be regenerated. Those that are affected by the ontology change need to be adjusted, either automatically or with some expert input, and then the KG does not need to be regenerated from scratch, potentially saving resources and, in turn, being more sustainable.

A further challenge that needs to be addressed is the distributed nature of the Semantic Web. Changes which need incorporating in the KG can come from the ontology or from the source data from which the KG is constructed. Both of these can be considered to be internal or external. Internal sources of change are easier to handle because of an existing direct communication channel. However, in real-world scenarios, the ontology or data used in a KG can be provided by different organisations with their own processes. With the separation between ontology engineering and KG construction, we enable organisations to adapt either one or both parts of the methodology. Further investigations are necessary in the future to identify pitfalls and challenges in such scenarios.

4 Conclusions and Future Work

In this short article, we give a high-level overview of the LOT4KG methodology. We present a first-of-its-kind theoretical methodology, which is based on previous work for dealing with the KG lifecycle as a whole. LOT4KG presents an extension to the LOT framework [6]: the inclusion of the KG lifecycle, describing the general steps that are followed when creating a KG from a given ontology or schema. In the future, we plan to make a lower-level definition of activities available, similar to what is already published for LOT. Furthermore, we discussed how ontology and KG evolution compare to the proposed methodology and how we plan to continue to extend it to make the lifecycle firmly encompass the maintenance of the artefacts as well, calling it LOT4KG in the future. This will lead to the definition of evolution activities on both levels, the ontology and KG. Implementations of the KG lifecycle are also of interest; however, these can be highly dependent on the available infrastructure. More interesting is the investigation into the evolution activities, as tool support is, to the best of our knowledge, still scarce.

This methodology is the first of its kind to combine the lifecycles of ontology and KG. The methodology can be beneficial in different real-world scenarios, especially when considering the sometimes vague separation between ontology/schema and KG. It would, therefore, enforce a clearer separation and would urge KG engineers to look at the ontology as an artifact by itself which requires its own evolution, separate from the KG. Further, with a clear separation between ontology KG, the methodology is also applicable in cases where the ontology and KG are not in the hands of the same person or team. The ontology can be engineered and maintained completely disconnected from the KG and vice versa. Once extended to include the evolution, LOT4KG will be helpful by providing tangible actions that need to be taken to keep the KG up to date when the ontology evolves. With the formalisation of such a methodology, we open up discussion on how ontologies and KGs are engineered today. The methodology also fosters further methodological research, as the Semantic Web community has to some extent mastered the engineering of ontologies and KGs but still needs to work on maintaining them over time.

Acknowledgments

María Poveda-Villalón is funded by the European Union's Horizon 2020 research and innovation programme under the grant agreement no. 101016854 (AURORAL). David Chaves-Fraga is funded by the Galician Ministry of Education, University and Professional Training and the European Regional Development Fund (ERDF/FEDER program) through grants ED431C2018/29 and ED431G2019/04. Diego Conde-Herreros is supported by the project *Knowledge*

Spaces (Grant PID2020-118274RB-I00 funded by MCIN/AEI/10.13039/50110 0011033) & by Spanish Statistical Office (INE).

References

1. Chaves-Fraga, D., Corcho, O., et al.: Systematic construction of knowledge graphs for research-performing organizations. Information **13**(12) (2022). https://doi.org/10.3390/info13120562
2. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series (1997)
3. Khattak, A.M., Latif, K., Lee, S.Y., Lee, Y.K., Rasheed, T.: Building an integrated framework for ontology evolution management. In: 12 th International Conference on International Business Information Management Association, Malaysia (2009)
4. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology (2001)
5. Polleres, A., Pernisch, R., et al.: How Does Knowledge Evolve in Open Knowledge Graphs? Transactions on Graph Data and Knowledge **1**(1), 11:1–11:59 (2023). https://doi.org/10.4230/TGDK.1.1.11
6. Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., García-Castro, R.: LOT: An industrial oriented ontology engineering framework. Engineering Applications of Artificial Intelligence **111**, 104755 (May 2022). https://doi.org/10.1016/j.engappai.2022.104755
7. Radulovic, F., Poveda-Villalón, M., Vila-Suero, D., Rodríguez-Doncel, V., García-Castro, R., Gómez-Pérez, A.: Guidelines for linked data generation and publication: An example in building energy consumption. Automation in Construction **57**, 178–187 (2015). https://doi.org/10.1016/j.autcon.2015.04.002
8. Simsek, U., Angele, K., Kärle, E., Opdenplatz, J., Sommer, D., Umbrich, J., Fensel, D.: Knowledge graph lifecycle: Building and maintaining knowledge graphs. In: Second International Workshop on Knowledge Graph Construction. vol. 2873. CEUR-ws.org (2021)
9. Stojanovic, L.: Methods and tools for ontology evolution. PhD Thesis, Karlsruhe Institute of Technology, Germany (2004)
10. Wardhana, H., Ashari, A., Kartika, A.: Review of Ontology Evolution Process. International Journal of Computer Applications **179**(25), 26–33 (Mar 2018). https://doi.org/10.5120/ijca2018916537
11. Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., Plexousakis, D., Sabou, M.: Ontology evolution: a process-centric survey. The Knowledge Engineering Review **30**(1), 45–75 (Jan 2015). https://doi.org/10.1017/S0269888913000349