Chapter 23
Multilingual Knowledge Systems as Linguistic Linked Open Data

Alena Vasilevich and Michael Wetzel

Abstract Creation and re-usability of language resources in accordance with Linked Data principles is a valuable asset in the modern data world. We describe the contributions made to extend the Linguistic Linked Open Data (LLOD) stack with a new resource, Coreon MKS, bringing together concept-oriented, language-agnostic terminology management and graph-based knowledge organisation. We dwell on our approach to mirroring of Coreon's original data structure to RDF and supplying it with a SPARQL endpoint. We integrate MKS into the existing ELG infrastructure, using it as a platform for making the published MKS discoverable and retrievable via a industry-standard interface. While we apply this approach to LLOD-ify Coreon MKS, it can also provide relevant input for standardisation bodies and interoperability communities, acting as a blueprint for similar integration activities.

1 Overview and Objectives of the Pilot Project

In a world depending on knowledge sharing, data-driven businesses and research communities are concerned with the creation, sharing, and use of language resources in accordance with Linked Data principles, which ensure better data discoverability, standardised structure, and cost savings for all parties involved in the creation of structured data. Robust, coherent, and multilingual information standards are needed to enable information exchange among public organisations, similar to standards that have been fostering technical interoperability for decades (Guijarro 2009).

We extend the Linguistic Linked Open Data (LLOD) stack with a new resource, Multilingual Knowledge System (MKS). MKS caters for the discovery, access, retrieval, and re-usability of terminologies and other interoperability assets organised in knowledge graphs (KG) in a taxonomic fashion. As a semantic knowledge repository, its main forte is the ability to exchange information among acting systems, ensuring that its precise *meaning* is understood and preserved among all parties, in any language. Injecting structure into the language data and expanding the result-

Alena Vasilevich · Michael Wetzel
Coreon GmbH, Germany, alena@coreon.com, michael@coreon.com

G. Rehm (ed.), *European Language Grid*, Cognitive Technologies,
https://doi.org/10.1007/978-3-031-17258-8_23

ing KG with multilingual terminologies, Coreon uses the European Language Grid (ELG) as a platform for making the published resources discoverable and retrievable through SPARQL, a protocol widely used for the retrieval of information from Semantic Web resources. While existing SPARQL tools enable users to query knowledge graphs, they are rarely used for termbases and other terminology resources, i. e., core data sources for translation and localisation (Stanković et al. 2014). This step makes Coreon integration into other systems tool-independent: instead of using the proprietary API, it relies on LLOD standards.

The goal of our contribution is to deliver MKS resources to the Semantic Web community, enabling it to query concept-oriented multilingual structured data with a well-established industry-standard syntax, and to promote the development of data multilingualism within the Semantic Web. In the long run, MKS as a LLOD resource can provide relevant input for standardisation bodies and interoperability communities: acting as a blueprint for similar integration activities, it can be viewed as a starting point for an international standard. We share our experience with ISO/TC37 SC3[1] working groups as a draft for a technical recommendation on how to represent TermBase eXchange (TBX) dialects as RDF.

2 Making Coreon Data Structure LLOD-compatible

Resource Description Framework (RDF) and Web Ontology Language (OWL) are standardised formats for representing Semantic Web data. They support data integration and offer a plethora of tools and methods for data access. SPARQL operates on RDF/OWL resources allowing users to retrieve structured responses to submitted queries. To express queries, it utilises triple patterns that are to be matched by RDF/OWL triples and filter conditions, imposing ranges for literals (Almendros-Jiménez and Becerra-Terón 2021). Despite the emerging interest in publishing terminological resources as linked data, the LLOD stack has not been heavily utilised for this purpose so far (Buono et al. 2020).

We implemented a solution for Coreon MKS, making termbases discoverable and accessible for LLOD systems (Chiarcos et al. 2013). Normally data owners deploy a technology like a RDF triple store for their terminology tool, often developing or setting up a tedious data-mirroring process. We go beyond the limits of RDF/knowledge graph editors, which tend to be good at relation modeling but have weaknesses when it comes to capturing linguistic information.

At the core of the MKS lies a language-independent KG. Unlike other popular solutions within terminology management, linking is performed *not* at the *term* but at the *concept* level; therefore, abstracting from terms, we can model structured knowledge for phenomena that reflect the non-deterministic nature of human language, such as word sense ambiguity, synonymy, and multilingualism. Linking *per concept* also ensures smooth maintenance of relations without additional data clutter:

[1] https://www.iso.org/committee/48136.html

relation edges are independent from labels, terms and their variants, and other metadata. Besides the mirroring process between the Coreon data model and an RDF graph, the RDF vocabulary was established, covering classes, relations, additional term-descriptive information, and administrative metadata. It binds elements into RDF triples. At this stage it was critical to identify information objects and mapping of predicates and literals.

```
{"created_at": "2021-04-20T13:04:59.816Z",
    "terms":[
            {"lang": "en",
             "value": "screen",
             "id": "607ed17b318e0c181786b549",
            "concept_id": "607ed17b318e0c181786b545",
            "properties": []},
            {"lang": "de",
             "value": "Bildschirm",
             "id": "607ed195318e0c181786b55e",
            "concept_id": "607ed17b318e0c181786b545",
            "properties": []}
    ],
     "id": "607ed17b318e0c181786b545"}
```

Listing 1 Excerpt of the Coreon data structure.

Listing 1 shows relevant lines within the original JSON data structure that represents the sample concept “screen”, with *concept* ID and individual *term* IDs and their values highlighted. To transform this data structure into an RDF graph, the concept and its two terms are bound together in statements, i. e., RDF triples. Each triple comprises a subject, a predicate and an object; in our case, the concept will act as the subject, the terms become objects and the required predicate is named `hasTerm`. The complete sample set of triples serialised in RDF/Turtle is provided in Listing 2, with highlighted lines 9-10 indicating that the resource with ID `606336dab4dbcf018ed99308` belongs to the OWL class *coreon:Concept* and contains a term with ID `606336dab4dbcf018ed99307`.

In RDF and LOD, data is stored in an atomic manner, with predicates and uniform resource identifiers (URIs) linking elements together. In our case, all instances represented as classes receive unique identifiers. Together with unique IDs, the namespace `coreon:` unambiguously identifies any given element, regardless of whether it is a concept, term, property or a concept relation. Table 1 lists our RDF vocabulary, derived from the original MKS data structure. During the Coreon-to-RDF conversion, there were obvious candidates for classes, like `Concept` and `Term`; yet mirroring descriptive information like `Definition` or `TermStatus` and mapping taxonomic and associative concept relations turned out to be challenging. For the predicates we had to specify what information can be used, defining `owl:range` and `owl:domain`;

```
coreon:607ed17b318e0c181786b547 a coreon:Edge;
  coreon:edgeSource coreon:606336dab4dbcf018ed99308;
  coreon:edgeTarget coreon:607ed17b318e0c181786b545;
  coreon:type "SUPERCONCEPT_OF" .

coreon:606336dab4dbcf018ed99307 a coreon:Term;
  coreon:value "peripheral device"@en .

coreon:606336dab4dbcf018ed99308 a coreon:Concept;
  coreon:hasTerm coreon:606336dab4dbcf018ed99307 .

coreon:607ed17b318e0c181786b545 a coreon:Concept;
  coreon:hasTerm coreon:607ed195318e0c181786b55e,
    coreon:607ed17b318e0c181786b549 .

coreon:607ed17b318e0c181786b549 a coreon:Term;
  coreon:value "screen"@en .

coreon:607ed195318e0c181786b55e a coreon:Term;
  coreon:value "Bildschirm"@de .
```

Listing 2 Triples serialised in RDF / Turtle

```
coreon:hasTerm
  rdf:type owl:ObjectProperty ;
  rdfs:comment "makes a term member of a concept" ;
  rdfs:domain coreon:Concept ;
  rdfs:label "has term" ;
  rdfs:range coreon:Term .
```

Listing 3 Specification of a predicate

e. g., the predicate `hasTerm` can only accept resources of type `coreon:Concept` as a subject (`owl:domain`). Listing 3 provides a full specification of this predicate.

	OWL Type	Coreon RDF Vocabulary
Classes	owl:Class	coreon:Admin, coreon:Edge, coreon:Concept, coreon:Flagset, coreon:Property, coreon:Term
Predicates	owl:ObjectProperty	coreon:hasAdmin, coreon:hasFlagset, coreon:hasProperty, coreon:hasTerm
Values	owl:AnnotationProperty	coreon:edgeSource, coreon:edgeTarget, coreon:id, coreon:name, coreon:type, coreon:value

Table 1 Derived Coreon RDF vocabulary

3 Real-Time Data Access via a SPARQL Endpoint

With the vocabulary defined, we equipped Coreon's export engine with a RDF publication mechanism, including the export in relevant syntax flavours (Turtle, N3, JSON-LD). The Coreon cloud service was supplied with a real-time accessible SPARQL endpoint via Apache Jena Fuseki.[2] It conforms to all published standards and tracks revisions and updates in the under-developed areas of the standard. Running as a secondary index in parallel with the repository's data store, Fuseki catches any changes made by data maintainers, updating the state of the repository in real time. Listing 4 demonstrates a sample SPARQL query over a MKS that deals with wine varieties: here, we want to return all terms, including the values of the *Usage* flag in case the terms have them.

```
SELECT ?t ?termvalue ?usagevalue
    WHERE { ?t rdf:type coreon:Term .
            ?t coreon:value ?termvalue .
            OPTIONAL {  ?t coreon:hasProperty ?p .
                        ?p coreon:key "Usage" .
                        ?p coreon:value ?usagevalue .
            }
    }
```

Listing 4 Sample SPARQL query over MKS

Table 2 shows a subset of the linked data structures returned by this query, i. e., a term's URI, its value, and usage recommendation if available.

[t]	termvalue	usagevalue
http://www.coreon.com/coreon-rdf#[…]8b8aa	Riesling	
http://www.coreon.com/coreon-rdf#[…]8b8bb	Cabernet Sauvignon	Preferred
http://www.coreon.com/coreon-rdf#[…]8b8be	CS	Not allowed
http://www.coreon.com/coreon-rdf#[…]8b8c2	Merlot	

Table 2 Results of the sample SPARQL query (Listing 4): returned grape varieties

4 Conclusions and Results of the Pilot Project

We developed a pipeline to make MKS resources LLOD-compatible, mapping Coreon data structure to RDF, conceiving the Coreon-RDF vocabulary and publishing MKS resources via ELG. Besides making the SPARQL endpoint available

[2] https://jena.apache.org

through ELG, we implemented a productised piece of software, providing TermBase eXchange-like terminology resources in the RDF and Semantic Web context; a set of demo repositories is accessible via the endpoint through ELG. Beyond establishing structural interoperability, the implemented interface bridges Coreon with other Semantic Web systems, enabling querying of elaborate multilingual terminologies. Our mirroring approach can act as a blueprint for similar conversion and integration activities, viewed as a starting point for an international standard. Deployed through ELG, Coreon's SPARQL interface enables the Semantic Web community to query rich heterogeneous MKS data with a familiar, industry-standard syntax, promoting data accessibility and contributing to the development of multilingual resources within the Semantic Web.

Acknowledgements The work described in this article has received funding from the EU project European Language Grid as one of its pilot projects.

References

Almendros-Jiménez, Jesús Manuel and Antonio Becerra-Terón (2021). "Discovery and diagnosis of wrong SPARQL queries with ontology and constraint reasoning". In: *Expert Systems with Applications* 165, p. 113772. DOI: 10.1016/j.eswa.2020.113772.

Buono, Maria Pia Di, Philipp Cimiano, Mohammad Fazleh Elahi, and Frank Grimm (2020). "Terme-à-LLOD: Simplifying the Conversion and Hosting of Terminological Resources as Linked Data". In: *Proc. of the 7th Workshop on Linked Data in Linguistics, LDL@LREC 2020, Marseille, France, May 2020*. Ed. by Maxim Ionov, John P. McCrae, Christian Chiarcos, Thierry Declerck, Julia Bosque-Gil, and Jorge Gracia. ELRA, pp. 28–35.

Chiarcos, Christian, Philipp Cimiano, Thierry Declerck, and John P. McCrae (2013). "Linguistic Linked Open Data. Introduction and Overview". In: *Proceedings of the 2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data*. Pisa, Italy: ACL, pp. i–xi.

Guijarro, Luis (2009). "Semantic interoperability in eGovernment initiatives". In: *Computer Standards & Interfaces* 31.1, pp. 174–180. DOI: 10.1016/j.csi.2007.11.011.

Stanković, Ranka, Ivan Obradović, and Miloš Utvić (2014). "Developing Termbases for Expert Terminology under the TBX Standard". In: *Natural Language Processing for Serbian-Resources and Applications*, pp. 12–26.