skos-history: Exploiting Web Standards for Change Tracking in Knowledge Organization Systems

Joachim Neubert
ZBW – Leibniz Information Centre for Economics, Kiel/Hamburg, Germany
Neuer Jungfernstieg 21
20254 Hamburg
j.neubert@zbw.eu

ABSTRACT

"What's new?" and "What has changed?" are questions users of knowledge organizations systems, such as thesauri, classifications or taxonomies, ask, when new versions of such vocabularies are published. Until recently, it had been difficult for the publishers to provide this information (normally resorting to custom change logging within maintenance applications), and almost impossible for anybody else. With the widespread acceptance of SKOS as the standard publication and exchange format, the situation has changed fundamentally: Exact deltas between two sets of RDF triples can be computed, and the differences can be organized in a meaningful way through SPARQL queries, taking advantage of regular SKOS structures. *skos-history*[1] combines a script for creating a "version store" with queries accessing this store to generate standard reports such as "added concepts" or "changed notations", or more subtle changes like concept splits, as well as aggregated statistics on certain change types, or a complete change history for a single concept across multiple versions. To allow for interactive sorting, filtering and downloading these reports, and for editing the queries in an IDE-like environment to adapt them to different vocabularies or user needs, other open source libraries are integrated.

Keywords:

Versioning, Revision history, Concept evolution, KOS, RDF, Named Graphs, Graph differences, SKOS, SPARQL

1. INTRODUCTION

In the past, vocabularies have been published in a variety of formats. Organizations using these vocabularies could not rely on standardized tools to compare different versions. Change information that came with new versions therefore often was sketchy, in plain text, or missing at all. This information could not flexibly be prepared in end-user friendly ways, aggregating changes, e.g., by subject area, or providing drill-downs to the concepts actually changed. Neither was such information machine-actable, in order to update data in downstream applications.

The picture changes as more and more vocabularies become available in SKOS format (a W3C recommendation since 2009) as RDF files. Multiple versions can be loaded into a triple store – thanks to the SPARQL 1.1 HTTP Update, Query and Graph Store protocols (W3C recommendations since 2013) in a generic, implementation-independent way.

WWW 2016 Companion, April 11-15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889304

The next section outlines the structures of a skos-history "version store". Section 3 describes the generation of change reports by generalized skos-history queries exploiting the commonalities of vocabularies in SKOS format. Section 4 shows how queries can be adapted to take advantage of the specifics of certain vocabularies, and Section 5 deals with real-world use cases of the software.

2. THE SKOS-HISTORY VERSION STORE

The loading of two or more files with versions of a SKOS vocabulary is described in the tutorial[2]. During the load process, named graphs for each version are created. Deltas between the versions are computed and saved as named graphs, too, split up in inserted and deleted triples. Metadata about the versions as well as the deltas is saved in a separate version history graph, making use of the Dataset versioning ontology (closely related to the ISO 25964 standard on thesauri and its correspondence to SKOS[1]) and the skos-history ontology.[3] The current version of the vocabulary and the chain of prior versions with their according deltas can be accessed by this metadata in a generic way.

The whole data load and preparation process is performed through scripted HTTP requests. That means that no proprietary data administration interfaces of the triple store implementations have to be addressed, and the script can be kept generic.

3. STANDARD QUERIES FOR CHANGE REPORTS

The delta graphs may comprise thousands – and in case of larger changes even hundreds of thousands – of triples. Yet, since knowledge organization systems and their expression in SKOS are generally modeled with concepts in focus, connected by a few well-known semantic relations, meaningful standard reports can be extracted from the version store. For this purpose, skos-history provides a set of queries, for example:

- Added and deleted (or deprecated) concepts – these is the most essential information for the human users of a vocabulary (such as subject indexers in a library)
- Changed preferred label or notation of concepts – this information often is highly relevant for downstream systems, which make use of the vocabulary
- Labels moved to added concepts – that may indicate a concept split

The queries use the current and the previous version as rewritable defaults for comparison. Whereas these queries list certain types of changes between two versions, a 'concept_deltas' query allows to trace all changes to a concept across multiple versions.

Further generic queries allow the inspection of the service graph of the version store and the version history graph, or give an overview of versions present in the store.

4. ADAPTION OF QUERIES

To be most useful, queries often can be adapted to specifics of a vocabulary at hand. Many vocabularies, e.g., have subdivisions, such as micro-thesauri or subject category schemes, which can be used to group changes in a more meaningful way than a plain alphabetical listing (see Figure 1).

The skos-history Github repository facilitates experimentation with the standard queries, and with further queries already adapted to particular vocabularies and published as examples. For this purpose, it employs SPARQL Lab, a wrapper around the YASQUE/YASR SPARQL client libraries[3] which supports the loading of queries from Github and the injection of variables (e.g., version identifiers) into queries via URL query arguments. Accordingly constructed links load a query into an IDE-like environment (provided by YASQE) ready for execution or for adaption to different user needs. A YASR-supported display of query results allows for merging URIs with their (pref)labels and for sorting or filtering, thus making the resulting reports suitable for end users.

5. REAL-WORLD USE

The skos-history approach has been developed in the course of the four-year overhaul of the STW Thesaurus for Economics[2], primarily to make the vast amount of changes transparent to the users of STW within and outside of ZBW. During a beta phase (v8.14), links to live-generated change reports, as described in section 4, were supplied[4]. For production (from v9.0), the results of the same queries were saved as JSON files, and are delivered without dependency to the availability of a SPARQL endpoint, which additionally allows to offer direct download links[5]. For an overview over the evolution of the vocabulary as a whole, visualizations with the aggregated numbers of insertions and deletions across multiple versions are provided, which allow for a drill-down from the sub-thesaurus level to the individual concept[6].

The National Library of Finland uses skos-history in a different scenario: They maintain the Finnish general thesaurus (YSA) and its Swedish language counterpart (Allärs) as well as the multilingual General Finnish Ontology (YSO). Allärs needs to follow the changes made to YSA, and YSO has to follow changes made in both YSA and Allärs. For the examination of changes, a user interface[7] has been created, which allows for the interactive selection of vocabularies, versions and type of changes, and hides the SPARQL queries completely from the user. Based on a skos-history version store, standard queries and additionally crafted ones (covering e.g. notes or skos concept groups) are used. Changes made to YSA and Allärs are automatically propagated to YSO as proposed changes and additions, using a set of scripts and SPARQL queries that rely on the skos-history version store.

6. CONCLUSION

Heavily relying on web standards-conforming third-party open source software and data publication, the skos-history project added and made available a relatively short script and a set of flexible and easy-to-adapt queries. This allows us - and others - to answer the user questions cited at the beginning of this paper in a way that was not achievable before.

7. ACKNOWLEDGEMENTS

Thanks to Osma Suominen for taking part in the skos-history development and for the description of its use at NatLibFi.

8. REFERENCES

[1] ISO TC46/SC9/WG8 working group and Antoine Isaac. Correspondence between ISO 25964 and SKOS/SKOS-XL Models. 2013. http://www.niso.org/schemas/iso25964/correspondencesSKOS/.

[2] Neubert, J. Leveraging SKOS to Trace the Overhaul of the STW Thesaurus for Economics. *Proc. Int'l Conf. on Dublin Core and Metadata Applications*, (2015).

[3] Rietveld, L. and Hoekstra, R. The YASGUI Family of SPARQL Clients. 2015. http://www.semantic-web-journal.net/content/yasgui-family-sparql-clients.

[1] https://github.com/jneubert/skos-history
[2] https://github.com/jneubert/skos-history/wiki/Tutorial
[3] https://github.com/jneubert/skos-history/wiki/Versions-and-Deltas-as-Named-Graphs
[4] http://zbw.eu/stw/version/8.14/changes
[5] http://zbw.eu/stw/version/9.0/changes
[6] http://zbw.eu/stw/version/9.0/relaunch
[7] http://ysa-to-yso.dev.finto.fi/

secondLevelCategory	deprecatedConcept	replacedByConcept
V.03 Macroeconomics	Asset accumulation	Saving incentives
V.03 Macroeconomics	Consumption statistics	Household survey
V.03 Macroeconomics	Household expenditure	Private consumption
V.03 Macroeconomics	Intertemporal income distribution	Intergenerational mobility
V.03 Macroeconomics	Macroeconomic effect	Impact assessment

Figure 1. Change report on deprecated concepts, adapted to STW Thesaurus for Economics.

The report is organized by the second-level of the STW subject category system to which the descriptors are attached. It includes hints to concepts which replace the deprecated ones. The deprecation of concepts (instead of their deletion) via owl:deprecated and the use of dcterms:isReplacedBy are not standardized by SKOS.

The links however, as in all skos-history change reports, use SKOS and Linked Data standard features: The URIs link to a human and machine readable representation of the concepts. The link texts stem from skos:prefLabel properties, in the language selected in the query.