Leveraging SKOS to Trace the Overhaul of the STW Thesaurus for Economics

Joachim Neubert
ZBW – Leibniz Information
Centre for Economics,
Germany
j.neubert@zbw.eu

Abstract

"What's new?" and "What has changed?" are questions users of Knowledge Organization Systems (KOS), such as thesauri or classifications, ask when a new version is published. Much more so, when a thesaurus existing since the 1990s has been completely revised, subject area for subject area. After four intermediately published versions in as many consecutive years, STW Thesaurus for Economics[1] has been re-launched recently in version 9.0. In total, 777 descriptors have been added; 1,052 (of about 6,000) have been deprecated and in their vast majority merged into others. More subtle changes include modified preferred labels, or merges and splits of existing concepts. We here describe how these changes were tracked, making use of the published SKOS (Miles & Bechhofer, 2009) files of the versions, loading them into named graphs of a SPARQL endpoint and executing queries on them. An ontology supporting version and delta description and query formulation is introduced. High-level visualizations of aggregated change data and drill-downs to the actual concepts are presented. We finish with an outlook to the *skos-history* project[2], which generalizes and extends the methodology to different knowledge organization systems.

Keywords: KOS; thesaurus; versioning; version history; Linked Open Data; Semantic Web; SPARQL; named graphs; service description

1. Use cases for change tracking

Vocabularies published on the web – particularly vocabularies shared under an open license like the Open Database License[3] used by STW – can be downloaded without notification of the publisher and may be in use in multiple places and scenarios. Only some of them are known to the publisher. So there is no way to know for a maintainer which of the changes made to a vocabulary may or may not break things down the line. Handling changes quietly within an organization, as it was a widespread practice for a long time, isn't an option any more.

Several use cases for change tracing have been identified within the skos-history project:[4]

1. support for human indexers for adapting their subject indexing practice to the new version – the classical use case
2. support for re-indexing large sets of documents, in an automatic, semi-automatic or manual fashion – vocabulary changes may require dealing with already indexed documents retrospectively
3. support for the maintenance of vocabulary mappings – new mapping targets may have occurred or already mapped concepts may have been deleted or deprecated

---

[1] http://zbw.eu/stw
[2] https://github.com/jneubert/skos-history
[3] http://opendatacommons.org/licenses/odbl/
[4] https://github.com/jneubert/skos-history/wiki/List-of-Use-Cases

4. support for the maintenance of derived vocabularies (e.g., a subset covering a special interest field, for the use within an independent organization)
5. support for vocabulary-based automated indexing applications
6. support for search applications

Most of these use cases involve applications for which a machine readable input format is highly desired. A standard case, which can be handled automatically, is the replacement of changed preferred labels or notations for display purposes. Similarly, for obsolete descriptors, which have been merged completely into others, the indexing of documents can be switched automatically – while the update of a mapping to another vocabulary may require intellectual verification. Other types of changes, particularly a split of concepts, may require a complete review of already indexed documents.

Due to the possibly large efforts required for a migration to the latest version, at any point in time multiple versions of a vocabulary will be in use concurrently by different institutions. One goal of the set of practices described below is to enable users of a vocabulary to calculate the impacts a version upgrade will have in their particular scenario.

Section 2 of this paper outlines the basic approach of STW to vocabulary versioning. The method to track changes based on Linked Data is introduced and discussed in section 3. Based on this method, section 4 presents reports for tracking different types of individual changes, mostly focused on the first use case described above. Section 5 demonstrates the use of visualizations of aggregated data to understand high-level changes of the whole vocabulary. Section 6 introduces a detailed history of single concepts. Section 7 provides an outlook to future work.

2. Basic vocabulary versioning approach

Maintenance of STW is done within a custom application. During rework of larger parts of the vocabulary it may be in an inconsistent state. When such parts – such as e.g. "Money and financial markets" or "Information and communication", which might span multiple sub-thesauri – were finished, a new version was published, bearing a version number (marked up as "owl:versionInfo") and a version date (marked up as "dct:issued"). The URIs of the concepts stay stable (Hillmann, Sutton, Phipps, & Laundry, 2006) – however, the web pages for the concepts, which include RDFa semantic markup, are created and published for each version anew, and their URLs bear version numbers and language tags[5]. The rdf/xml and turtle expressions of a concept are versioned too. (Neubert, 2009) All files of previous versions remain accessible without limitations. As we observed that users often save the webpage address instead of the persistent URI as link, since 2010 the URI part "latest" instead of a version number has been serving as a default and redundant "symbolic link" to the latest version. All web pages of previous versions carry transparent "water-marks" to indicate their outdated status.

Once introduced, concepts are never deleted. Instead, obsolete concepts are stripped of all semantic relations, and are marked with a property "owl:deprecated true". A textual hint such as "Deprecated (last used in version 8.04)" is added as a "skos:historyNote" (in HTML together with a link to the according page). When applying, a "dct:isReplacedBy" property links to a still existing concept into which the deprecated concept was merged.[6]

Since its start on the web in 2009, STW has published lists of changes in the form of plaintext files. Additionally, as the RDFa-enhanced web pages and the bulk SKOS downloads of every published version have been kept available, users had a chance to look up changes and compare versions of concepts manually. Unfortunately, the plaintext change lists were not linked to the actual concept pages. Furthermore, there was no way of filtering nor aggregating the information – let alone accessing it for any kind of machine processing.

---

[5] e.g., http://zbw.eu/stw/version/8.14/descriptor/11716-2/about.en.html is the English version 8.14 page about the concept http://zbw.eu/stw/descriptor/11716-2 "Infrastructure".

[6] e.g., http://zbw.eu/stw/descriptor/12257-3

This clearly was not well-suited to the large-scale changes going on within the vocabulary. One option would have been extending the custom maintenance application to log actions and produce more expressive change reports. Yet, an application-specific hard-coded solution would have made it difficult to experiment with different forms of change reports. And whatever the outcome, it would have served STW only.

3. Using SKOS files for version comparisons

Instead of logging change transactions as they occur during the maintenance process, we decided to compare the final SKOS files of differing versions. This means that the methodology described here should be applicable not only by producers, but also by consumers of vocabularies or interested third-parties, without the need of out-of-band knowledge buried in whatever internal vocabulary maintenance system.

The approach is founded on an RDF database of vocabulary versions and computed version deltas (as described below), which can be flexibly evaluated by SPARQL queries. These can take advantage of the quite regular and predictable structure of SKOS files (as opposed to arbitrary ontologies). The database, referenced as "version store" in the remainder of the paper, is based on RDF named graphs[7] and created by the following steps:

1. load every version into a named graph
2. compute the delta between two versions and add it as two separate named graphs of insertions and deletions
3. add metadata describing and linking versions and deltas in a separate version history graph

For step 1, every triple/quad store which can deal properly with named graphs should suit – for STW we used Fuseki from the Apache Jena project. An additional experiment was conducted successfully with Sesame.

Step 2 can be approached by simply diff-ing version files in ntriples format by means of the operating system and splitting the result into insertions and deletions, or by creating the graphs directly in the version store by executing SPARQL update queries which ask for the triples of one version graph MINUS the triples of the other version graph. Neither method works well with blank nodes – they are perceived as deleted and inserted completely. However, their use should not be essential for any SKOS vocabulary, and if occurring (as in STW, where a few complex "use instead" notes were expressed using blank nodes), it makes most sense to filter them out.

[7] http://patterns.dataincubator.org/book/named-graphs.html

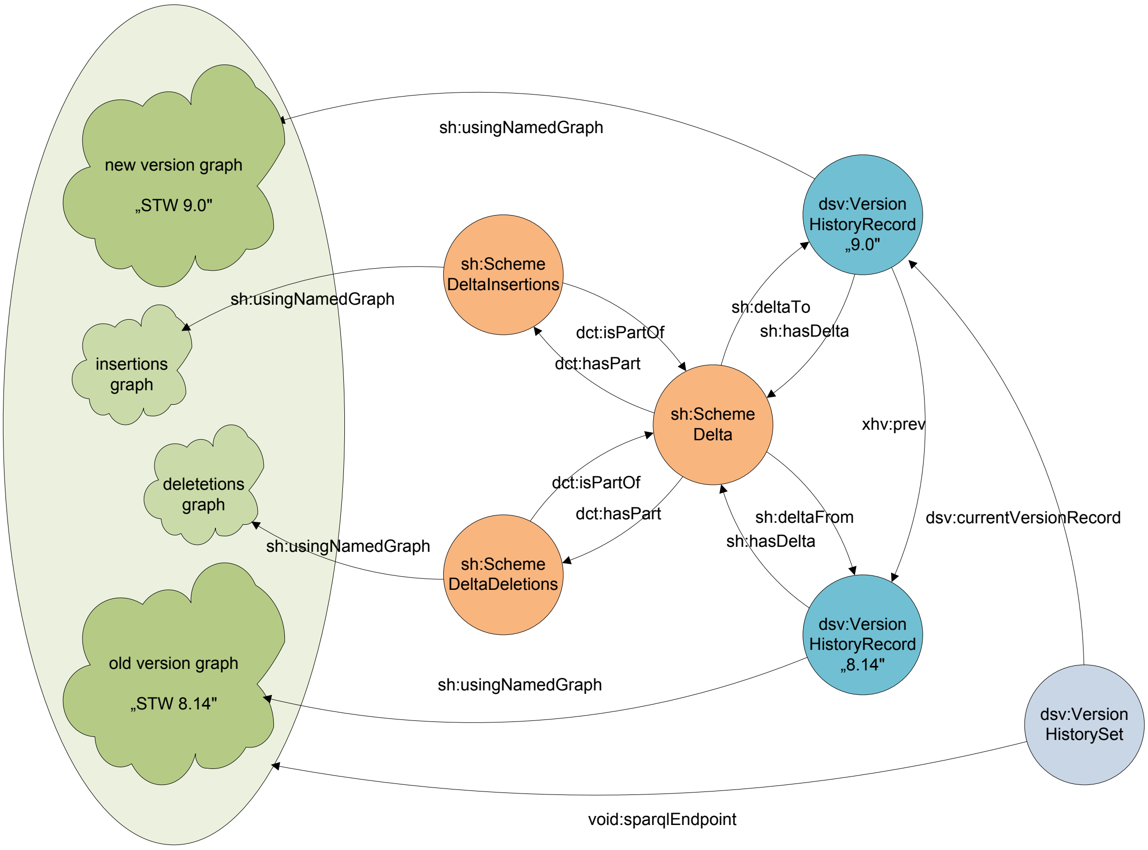

FIG. 1. Example data of the STW version history graph

For step 3, illustrated in FIG. 1, the emerging Dataset Versioning[8] ontology (dsv) was used for describing the versions, furthermore the SPARQL 1.1 Service Description[9] ontology (sd) for the named graphs and the newly developed skos-history[10] ontology (sh) for the deltas and additional plumbing, supplemented by Dublin Core elements (dc) and terms (dct), the Vocabulary of Interlinked Datasets (void) and XHTML (xhv).

The basic idea of Dataset Versioning (derived from the ISO 25964 data model) is providing a "dsv:VersionHistoryRecord" (VHR) identified by a "dc:identifier" plus an optional "dc:date" for each version. Each VHR links to the single "dsv:VersionHistorySet" (these links are omitted in FIG.1), which in turn points back to exactly one VHR through a "dsv:currentVersionRecord" property. (De Smedt, Vrang, & Papantoniou, 2015) Pairs of VHR are connected via a "sh:SchemeDelta", the chain of consecutive VHR is represented by "xhv:prev" links. The distinction between insertions and deletions parts, which link to the respective parts and named graphs, is implemented by different RDF classes.

For reference and re-use, all three steps for creating the version store are packaged in a publicly available bash script.[11] The version history set of STW is discoverable via a fix URI[12], as suggested in (ISO TC46/SC9/WG8 & Isaac, 2012). The embedded RDFa of the respective web page hints to the SPARQL endpoint containing the version store, which in turn includes the version history graph and the (default) service graph describing its overall structure.

[8] http://purl.org/iso25964/DataSet/Versioning#
[9] http://www.w3.org/ns/sparql-service-description#
[10] http://purl.org/skos-history/
[11] https://github.com/jneubert/skos-history/blob/master/bin/load_versions.sh. The repository version referenced in this paper is tagged STW_9.0
[12] http://zbw.eu/stw/version

The method and metadata structure described here can be applied to every set of versions of a SKOS vocabulary – provided the versions are available as separate files and bear some kind of identifier. However, it is not well suited when vocabulary changes are published as a stream of update events, such as the subject updates published by the Library of Congress Linked Data Service as ATOM feed[13]. In case of LCSH it should be possible, however, to fall back to time-stamped download files of the whole data set.

4. Tracking version changes in change reports

The following change reports have been developed based on change categories proposed in (Pessala et al., 2011) and subsequently enhanced within the skos-history project.[14] They operate on the version store and the metadata described in chapter 3. The queries used to generate the reports which are discussed in this chapter are publicly accessible.[15] In a "SPARQL Lab" environment (Neubert, 2014), they can be loaded from GitHub and executed, but also inspected and modified by users.

Since the execution of the queries may take more than ten seconds, the results are cached as machine-readable JSON files (for which "raw data" download links are offered, too). The YASR[16] library for Javascript supports formatted display of raw SPARQL results in browsers (Rietveld & Hoekstra, 2015). It provides additional user-friendly functionality, in particular the merge of concept URIs with their respective labels, presented as clickable links to the concepts (as shown in FIG. 2), furthermore paging for large result sets, and a quick search / filtering of the resulting tables.

4.1 Added and deprecated concepts

Added descriptors: This is the most basic information a report on vocabulary changes has to deliver. It can be obtained by asking for inserted concepts (identified by the occurrence of a "skos:prefLabel" triple in the insertions graph), for which no triples exist in the old version graph. The following SPARQL query can be executed against the public STW version store[17]:

```
# Identify concepts inserted with a certain version
#
SELECT distinct ?concept ?prefLabel
WHERE {
  # query the version history graph to get a delta and via that the relevant graphs
  GRAPH <http://zbw.eu/stw/version> {
    ?delta a sh:SchemeDelta ;
      sh:deltaFrom/dc:identifier "8.14" ;
      sh:deltaTo/dc:identifier "9.0" ;
      sh:deltaFrom/sh:usingNamedGraph/sd:name ?oldVersionGraph ;
      dct:hasPart ?insertions .
    ?insertions a sh:SchemeDeltaInsertions ;
      sh:usingNamedGraph/sd:name ?insertionsGraph .
  }
  # for each inserted concept, a newly inserted prefLabel must exist ...
  GRAPH ?insertionsGraph {
    ?concept skos:prefLabel ?prefLabel
  }
  # ... and the concept must not exist in the old version
  FILTER NOT EXISTS {
    GRAPH ?oldVersionGraph {
      ?concept ?p []
    }
  }
}
```

[13] http://id.loc.gov/techcenter/
[14] https://github.com/jneubert/skos-history/wiki/List-of-Change-Categories
[15] https://github.com/jneubert/skos-history/tree/master/sparql/stw
[16] http://yasr.yasgui.org/
[17] http://zbw.eu/beta/sparql/stwv/query

This (simplified) query gives a list of all added concepts with their preferred labels in all languages. For use on the STW web site, the query is extended in several ways: The language is restricted to that of the current user interface, the concepts are restricted to descriptors (excluding subject categories – see below), and an additional column is added to provide information about the subject area to which the descriptor was added. Furthermore, the extended queries allow external parametrization via VALUES clauses and provide reasonable defaults (e.g.: compare the latest and the penultimate version of a vocabulary).

To this end, the overall structure of STW is exploited: Besides the actual descriptors with their poly-hierarchical broader/narrower relationships, it provides a mono-hierarchical system of subject categories ("concept groups" in terms of ISO 25964), which forms the sub-thesauri of STW and bear notations.[18] Each descriptor is attached to one or more subject categories. For the majority of the change reports, the second level of this category system – e.g., "V.13 Labour" or "B.07 Marketing" – proved instrumental for breaking down the wide field of economics into about 80 meaningful subject areas.

Deprecated descriptors (with replacements): In a similar way, the deprecated descriptors are retrieved from the version store. This produces a table, which should be helpful for human indexers, but might also be used by scripts to update an already indexed database of documents.

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

FIG. 2. Change report: Deprecated descriptors (extract)

Added subject categories / Deprecated subject categories (with replacements): Similar to the reports for descriptors, these track changes of STW's category system. On a more global level, these reports expose where new fields of knowledge have emerged, or where on the contrary subdivisions are no longer regarded as necessary. For example, the subject category "W.19 Computer Software and Services Industries" (in version 8.04: "Data Processing") was renamed to "W.19 ICT industry" and extended with further sub-ordinated categories, namely "W.19.3 Broadcasting Industry", "W.19.4 Telecommunications" and "W.19.5 Information Services". These new subject categories cluster already existing descriptors scattered over the category system before as well as newly introduced ones complementing the newly formed field of knowledge.

4.2 Label changes

Changed preferred labels: Since SKOS requires at most one preferred label per concept per language, we can safely identify cases where this label has changed. In this report, we offer links to the old and the new version of the descriptor, by constructing a version-specific URL to the corresponding web pages. This allows users to directly compare these pages.

Since for STW subject categories the preferred labels are created by prepending the label itself with the notation of the category, the changed preferred labels report for subject categories reflect

[18] In SKOS, STW descriptors and subject categories are represented as subtypes of skos:Concept, namely zbwext:Descriptor and zbwext:Thsys (zbwext: http://zbw.eu/namespaces/zbw-extensions/)

also changes in the notation. This can reveal far-reaching changes in the STW category system, in the hierarchy or even in the assignment of partial category trees to sub-thesauri.

Added labels / Deleted labels: The former report shows all inserted preferred and alternate labels for descriptors (with the concept itself and its preferred label), the latter the deleted labels. Since labels do not carry an own identity and only the lexical values can be tracked, even minor changes in spelling (e.g., from “Advertising Industry” to “Advertising industry”) show up independently in the deletions and in the insertions list.

4.3 Hierarchical relations

Added narrower relationships / Added broader relationships: Changes in the descriptor hierarchy may be relevant in particular for newly inserted narrower concepts, so if the concepts to which the relationships were inserted are new, it is marked in these reports, too. Prior intermediate concepts in the hierarchy, which had been removed, are indicated also.

4.4 Other types of merges and splits

Splits: Labels moved to new descriptors: When a label is attached to another concept in a new version, this can be regarded as a hint that possibly the scope of the originating descriptor has changed. Particularly when the concept to which the label is attached is inserted with the same version, this may reveal a split of concepts (“Confidence interval”, for example, has been moved from “Estimation theory” to the newly introduced concept “Interval estimation”). This report, together with the Added-narrower one, can be taken as a basis for intellectual review of already indexed documents, which may or may not match the newly introduced narrower concept.

Merges: Labels moved from deprecated and split-up descriptors: This report covers the opposite situation, and particularly the case where the labels from a deprecated concept now are attached to other concepts than the one it was replaced by (e.g., the label “Royalties” was moved from the deprecated concept “Right of use” to “Charges”, while “Right of use” otherwise was merged into “Industrial property rights”). This can be taken as a hint that it may not be advisable to automatically re-index documents with the “replacedBy” concept without prior intellectual review.

Especially the two latest reports document a shift in the meaning of the remaining concepts, while their URIs have stayed the same. This may seem improper from a purely ontological point of view. However: In a real life environment defined by limited resources and cumbersome legacy library systems, nobody would want to take the additional effort to change already existing descriptors and re-index large amounts of documents without inescapable necessity.

Of course, the amount of necessary (re-)indexing is a factor which already has been taken into account up-front. Descriptors which have been used only a few times over the years are natural candidates for deprecation, while a re-indexing of thousands or ten-thousands of documents would not be considered with levity. Within ZBW, the holdings and the number of documents indexed with a particular descriptor are known, and the re-indexing can take place in parallel to the preparation of a new STW version. External parties using the thesaurus for indexing have to catch up afterwards. To provide information about the changes in an easily comprehensible way, to allow the estimation of subsequent efforts in local systems unknown to the editors of STW, is a primary goal of the approach described here.

5. Visualizing change with aggregated data

The change reports, as outlined above, allow tracking every single change from a certain class of changes. However, they are too detailed to facilitate insight into the development of the vocabulary as a whole. Yet, SPARQL 1.1 provides the means to query the version store for aggregated data. This allows the creation of statistics and charts which give a high-level overview over the changes to STW, particularly when aggregated over the complete amount of changes from version 8.06 to 9.0.

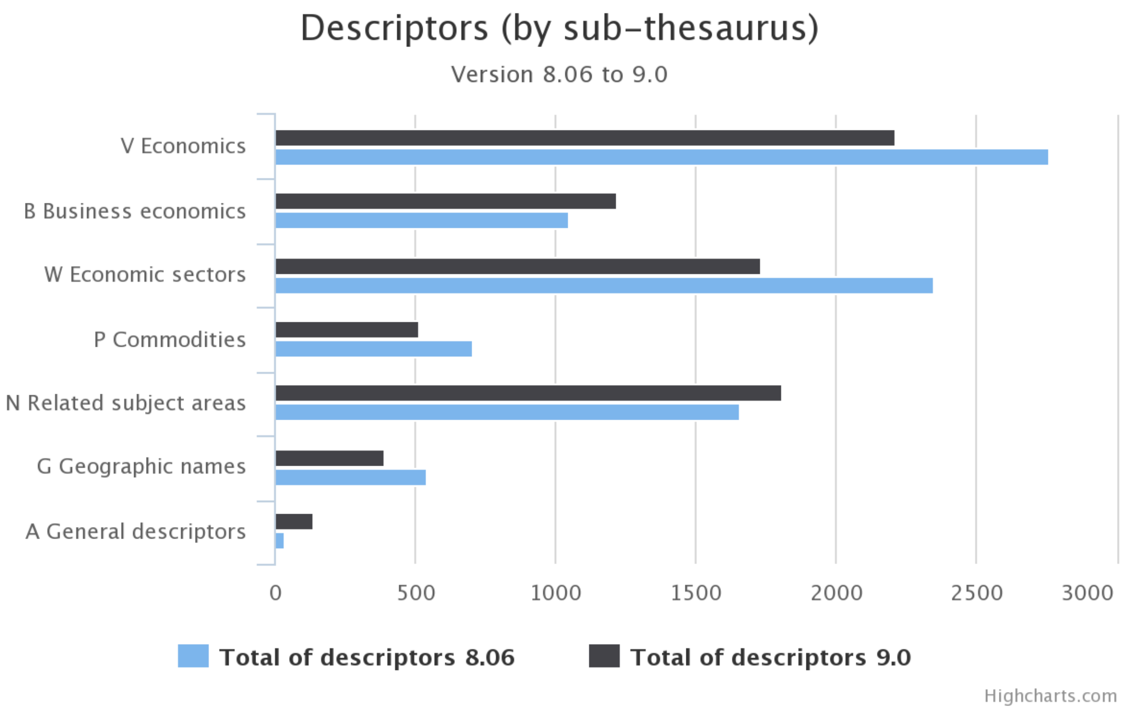

FIG. 3 Total number of descriptors

When we compare the total number of descriptors of these versions by sub-thesaurus (FIG. 3), we can see that the "Business economics" sub-thesaurus has been extended. The interactive graphics[19] allow drilling down and discovering that the number of descriptors has grown particularly for the subject categories "Management and business organization", "Logistics" and "Marketing". The "Economic sectors" and "Commodities" sub-thesauri both have decreased counts of active descriptors. This is true for the general "Economics" sub-thesaurus, too. However, a more detailed analysis, facilitated by the "Changed preferred labels/notations" report on subject categories, reveals that this is partly caused by the movement of the whole field of mathematical and statistical methods from "Economics" to "Related subject areas". On the whole, the branches of STW look more balanced after the overhaul.

While the overview charts give the net amount of descriptors, a series of more detailed charts (FIG. 4 and FIG. 5) shows the number of additions and deprecations within a certain part. These graphics support drill-downs, too.

In the example from the "Business economics" sub-thesaurus we can easily see that specifically in the fields of "Accounting" and "Corporate tax management" the deprecated/merged descriptors by far outnumber the added ones, and that a relatively small field such as "Operations research" has extended its coverage considerably.

The change graphics not only provide a high level overview. They work at the same time as a navigation tool, which allows focusing on the most interesting fields of change, and drill down into the change reports for added or deprecated concepts, by a passed-in search filter restricted to a particular 2nd level subject category. As the change reports link to the concepts themselves, this allows investigations up to the finest details.

[19] linked from http://zbw.eu/stw/relaunch

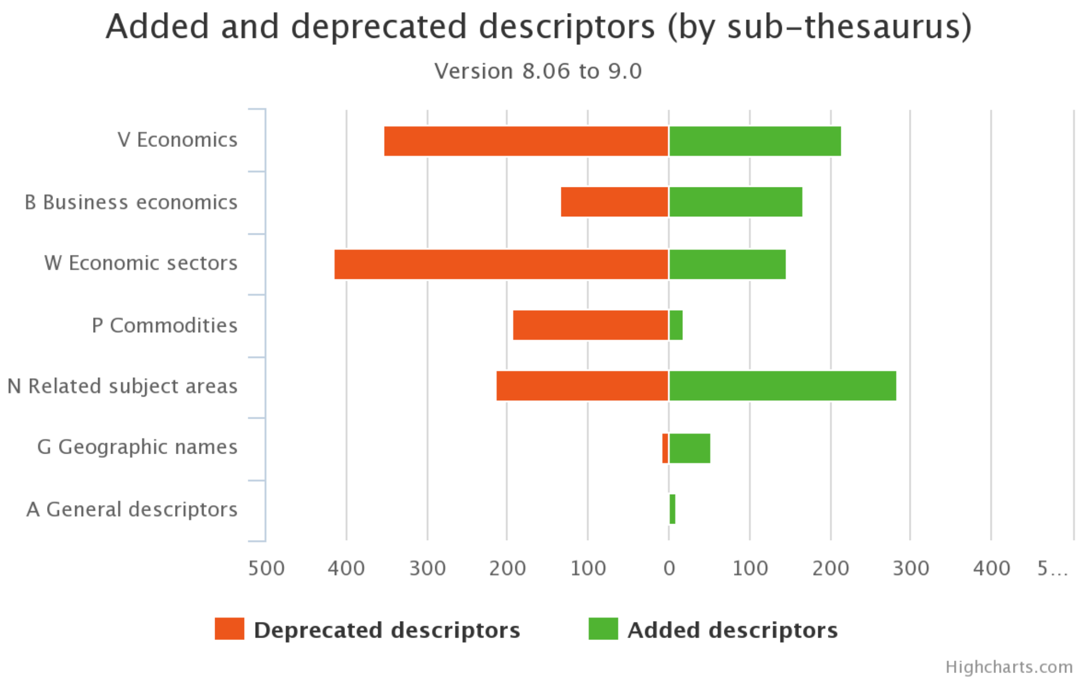

FIG. 4 Number of added and deprecated descriptors by sub-thesaurus, with drill-downs

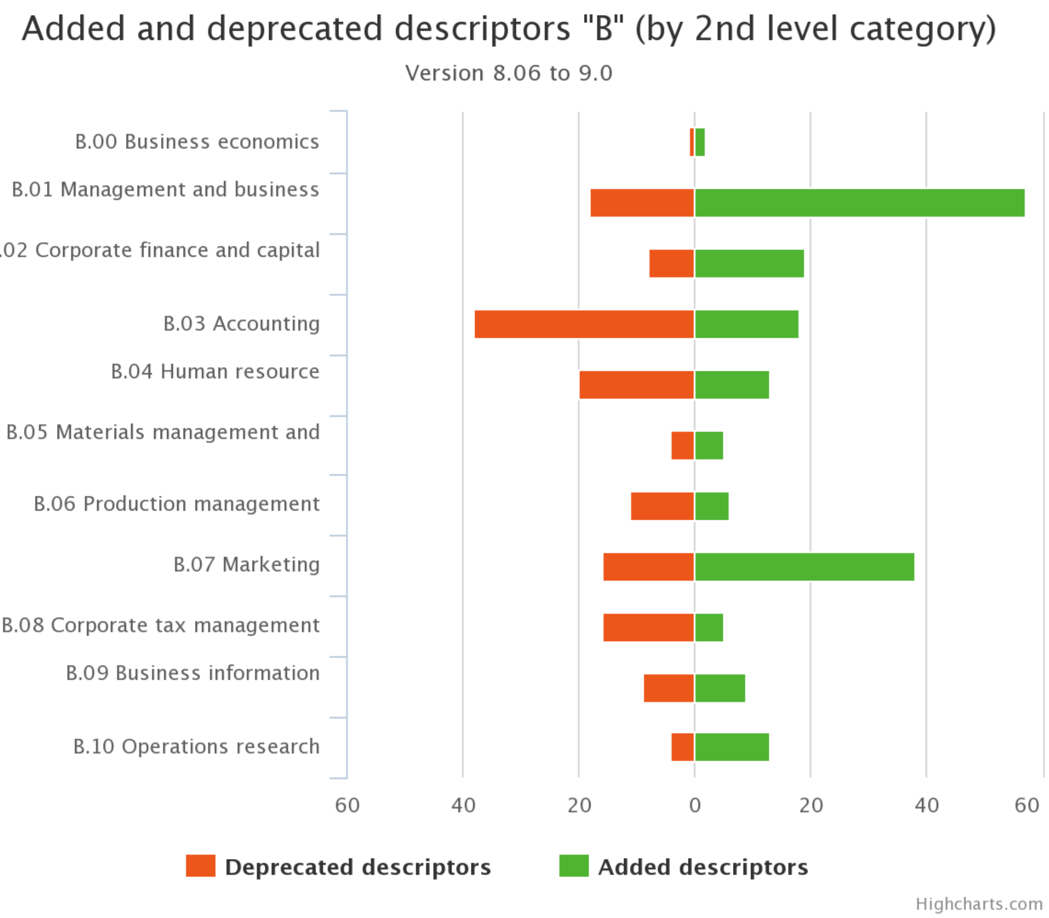

FIG. 5 Sub-thesaurus Business economics: Added and deprecated descriptors by 2nd level category, with drilldowns

6. Exposing the complete history of single concepts

From the point where a single concept is viewed, it would be very useful to be able to obtain the full history of that particular concept. To this end, a SPARQL construct query on the data of the version store has been developed[20]. It focusses on a single RDF subject and builds a temporary RDF graph, grouping all triple insertions and deletions for this subject by version delta. The query makes use of the Delta ontology as introduced by (Berners-Lee & Connolly, 2009). That allows us to track changes in (preferred or alternate) labels as well as changes in the relationships structure. Preliminary "Concept history (RDF/Turtle)" links to this graph are included on the web pages for all STW concepts since version 9.0.[21] It is planned to transform the concept version graph to a formatted web page with human-readable labels for the concepts in one of the upcoming versions of STW.

7. Future work

Besides providing a human-readable concept history, the emphasis for future work lies in the field of extending the described methodology to other SKOS vocabularies, and to probe and test it in various use case scenarios.

The methodology described in this paper is intended to work with any published SKOS vocabulary, without the need for out-of-band knowledge sealed in its maintenance environment and processes. First results show that it could be applied and worked for the Thesaurus for the Social Sciences (TheSoz), which differs from STW in the use of SKOS-XL labels. Further experiments are under way with the Finnish General Ontology (YSO), which makes heavy use of SKOS collections, and with the Agrovoc thesaurus maintained by FAO and available in multiple languages, which differ largely in coverage. We can assume that adapting to the different specifics of individual thesauri will reveal commonalities as well as fields where additional restrictions or extensions will prove necessary, as it has been shown above for the descriptor and subject category sub-types of concepts within STW.

The reports described in section 4 are currently intended and optimized for mostly human consumption. While they are provided in machine-readable JSON format, further work is required to evaluate their use in (semi-) automatic processing scenarios as described in the first chapter of this paper. This will reveal ways to bundle the data which are better suited to both fully automated update tasks as well as roll out of changes which require human judgment and are poorly supported by maintenance tool chains at the moment. Publication in "raw" RDF may be useful for merging data. For example, the set of concepts which had been split up could be merged with the number of times these concepts had been used for indexing local documents, in order to estimate the impact of a version update. Or the labels which have been moved to a new concept can be searched automatically within the titles or abstracts of all documents indexed with the split-up concept, in order to generate a list of suggestions for a semi-automatic re-indexing workflow.

The skos-history project should be instrumental in gathering information about differences in thesaurus and classification structures and different usage scenarios, in order to develop a set of tools and best practices to trace change in knowledge organization systems.

[20] https://github.com/jneubert/skos-history/blob/master/sparql/concept_deltas.rq
[21] e.g., http://zbw.eu/stw/version/9.0/descriptor/16269-4/about

Acknowledgements

With thanks to Sini Pessala (National Library of Finland) and Johan De Smedt (Tenforce, Belgium) for the cooperation and the discussions in the context of the skos-history project, and to Manuela Gastmeyer (ex-ZBW), who has built and maintained STW over more than two decades, including the overhaul traced in this paper.

References

Berners-Lee, T., & Connolly, D. (2009, August 27). Delta: an ontology for the distribution of differences between RDF graphs. RDF Diff, Patch, Update, and Sync -- Design Issues. Retrieved September 2, 2013, from http://www.w3.org/DesignIssues/Diff

De Smedt, J., Vrang, M. le, & Papantoniou, A. (2015). ESCO: Towards a Semantic Web for the European Labour Market. Accepted for the WWW Workshop Linked Data on the Web, Florence, Italy.

Hillmann, D. I., Sutton, S. A., Phipps, J., & Laundry, R. (2006). A Metadata Registry from Vocabularies UP: The NSDL Registry Project. *arXiv:cs/0605111*. Retrieved from http://arxiv.org/abs/cs/0605111

ISO TC46/SC9/WG8, & Isaac, A. (2012, June). Correspondence between ISO 25964 and SKOS/SKOS-XL Models. Retrieved from http://www.niso.org/apps/group_public/download.php/9627/Correspondence%20ISO25964-SKOSXL-MADS-2012-10-21.pdf

Miles, A., & Bechhofer, S. (2009, August 18). SKOS Simple Knowledge Organization System Reference. Retrieved March 23, 2015, from http://www.w3.org/TR/2009/REC-skos-reference-20090818/

Neubert, J. (2009). Bringing the "Thesaurus for Economics" on to the Web of Linked Data. In *Proc. WWW Workshop on Linked Data on the Web (LDOW 2009), Madrid, Spain*. Retrieved from http://ceur-ws.org/Vol-538/ldow2009_paper7.pdf

Neubert, J. (2014, November 14). Publishing SPARQL queries live. Retrieved from http://zbw.eu/labs/en/blog/publishing-sparql-queries-live

Pessala, S., Seppälä, K., Suominen, O., Frosterus, M., Tuominen, J., & Hyvönen, E. (2011). MUTU: An Analysis Tool for Maintaining a System of Hierarchically Linked Ontologies. In *ISWC 2011 - Ontologies come of Age Workshop (OCAS)*. Bonn, Germany. Retrieved from http://www.seco.tkk.fi/publications/2011/pessala-et-al-mutu-2011.pdf

Rietveld, L., & Hoekstra, R. (2015, under review). The YASGUI Family of SPARQL Clients. Retrieved from http://www.semantic-web-journal.net/content/yasgui-family-sparql-clients