Improving the Quality of SKOS Vocabularies with Skosify

Osma Suominen and Eero Hyvönen

Semantic Computing Research Group (SeCo),
Aalto University, Department of Media Technology
University of Helsinki, Department of Computer Science
firstname.lastname@aalto.fi, http://www.seco.tkk.fi/

Abstract. Simple Knowledge Organization System (SKOS) vocabularies are commonly used to represent lightweight conceptual vocabularies such as taxonomies, classifications and thesauri on the Web of Data. We identified 11 criteria for evaluating the validity and quality of SKOS vocabularies. We then analyzed 14 such vocabularies against the identified criteria and found most of them to contain structural errors. Our tool, **Skosify**, can be used to automatically validate SKOS vocabularies and correct many problems, helping to improve their quality and validity.

1 Introduction

Controlled vocabularies such as taxonomies, classifications, subject headings and thesauri [3] have been used for more than a century in the classification of books, music and other documents. In the Web era, the use of such vocabularies has greatly expanded into the description of personal interests, news categories, blog topics, art and literary styles, music genres and many other semantic attributes.

Such vocabularies are increasingly being published using the Simple Knowledge Organization System (SKOS) standard of describing vocabularies by means of RDF structures [14]. As an example, many library classifications have been published as SKOS vocabularies, allowing various library catalogs using those classifications to be published as Linked Data and then easily integrated using RDF tools [7,13,19], enabling applications such as semantic information retrieval over multiple datasets [8], query expansion and recommendation [16].

However, the benefits of SKOS in data integration are only realizable if the SKOS vocabulary data is structurally valid and makes use of the SKOS entities in a meaningful way. To this end, the SKOS reference [14] defines a number of *integrity conditions* that can be used to detect inconsistencies in a SKOS vocabulary. In addition, *validation tools* are available for verifying that a SKOS vocabulary follows generally accepted best practices for controlled vocabularies which have not been codified in the SKOS reference. Many SKOS vocabularies are currently published by automatically converting vocabularies from legacy formats into SKOS. Structural problems in the resulting SKOS files may be difficult to notice for vocabulary publishers, but may cause problems for users of the vocabularies.

In this study, we have surveyed the current quality and validity of published SKOS vocabularies and found many examples of structural problems. Our intent is to improve on the current state of standards compliance and quality of SKOS vocabularies published as RDF or Linked Data. We therefore seek answers for the following **research questions**:

1. What criteria can be used to validate a SKOS vocabulary?
2. How well do existing SKOS vocabularies fulfill those criteria?
3. How can SKOS vocabulary quality be improved?

To answer the first research question, we compiled a list of suitable validation criteria for SKOS vocabularies, detailed in Section 3. To answer the second research question, we used an online validation tool to validate thirteen SKOS vocabularies found on the web as well as one vocabulary produced in our own research group. This validation is detailed in Section 4. To answer the third research question, we created the Skosify tool to find and correct many kinds of inconsistencies and problems in SKOS vocabularies. The tool and the problems it can be used to correct are further described in Section 5.

2 Related Work

The SKOS reference specifies a number of *integrity conditions* which must be fulfilled for the vocabulary to be considered valid [14]. Many of these conditions are based on earlier standards for structuring controlled vocabularies and thesauri, including ISO 2788 [1] and the British standard BS8723 Part 2 [2]. These conditions may be considered a minimum set of validation and/or quality criteria for SKOS vocabularies; there are also many vocabulary-related best practices which go beyond the integrity conditions codified in SKOS. A more thorough set of quality criteria (hereafter known as the **qSKOS criteria**) for SKOS vocabularies [12] and a validation tool that can be used to measure vocabularies against these criteria has been developed in the **qSKOS project**[1]. On a more theoretical level, Nagy et al. have explored the various structural requirements of SKOS vocabularies in different application scenarios [16].

The **PoolParty online SKOS Consistency Checker**[2] (hereafter known as the **PoolParty checker**) performs many checks on SKOS vocabularies, including the SKOS integrity conditions. It also indicates whether the vocabulary can be imported into the online **PoolParty thesaurus editor** [18]. The W3C used to host a similar online SKOS validation service, but it was not kept up to date with the evolution of SKOS, and is no longer available.

More general validation services for RDF and Linked Data have also been developed. The **W3C RDF Validation Service**[3] can be used to verify the syntax of RDF documents. The **Vapour** [5] and **RDF:Alerts** [10] systems are

[1] `https://github.com/cmader/qSKOS/`
[2] `http://demo.semantic-web.at:8080/SkosServices/check`
[3] `http://www.w3.org/RDF/Validator/`

online validation tools intended to spot problems in Linked Data. A recent and thorough survey of general RDF and Linked Data validation tools is given in [10]; however, to our knowledge, none of these tools have any specific support for SKOS vocabularies.

3 Validation Criteria

In order to find a suitable set of criteria for checking the validity and quality of SKOS vocabularies, we compared three lists of such criteria: the integrity conditions defined in the SKOS reference [14], the validity checks performed by the PoolParty checker, and the list of quality criteria from the qSKOS project. The results of matching these criteria are shown in Table 1. The last column of the table indicates whether our tool, Skosify, can detect and correct each issue, a feature discussed in more detail in Section 5.

3.1 Selection of Criteria

We excluded several qSKOS criteria that were very context-specific and/or would be difficult to improve algorithmically. For example, the degree of external links (VQC4) and language coverage (VQC11b) in a vocabulary can be measured, but the results have to be interpreted in the context of the intended use of the vocabulary and would not be easy to improve using a computer program.

The different lists of criteria use varying levels of granularity. For example, the PoolParty checker considers both the existence of more than one `prefLabel`[4] per language and the use of disjoint label properties as violating the Consistent Use of Labels check, while qSKOS has a single criterion (VCQ8 *SKOS semantic consistency*) which encompasses all kinds of violations of the SKOS integrity

[4] Typographical note: words set in typewriter style that don't include a namespace prefix, such as `Concept` and `prefLabel`, refer to terms defined by SKOS [14].

Table 1. Comparison of Validation Criteria for SKOS Vocabularies.

Criterion name	SKOS	PoolParty checker	qSKOS	Skosify
Valid URIs	-	Valid URIs	(VQC5 is stricter)	not checked
Missing Language Tags	-	Missing Language Tags	VQC11a	corrected
Missing Labels	-	Missing Labels	VQC10	corrected partially
Loose Concepts	-	Loose Concepts	(VQC1 is stricter)	corrected
Disjoint OWL Classes	S9, S37	Disjoint OWL Classes	(VQC8)	corrected partially
Ambiguous `prefLabel` values	S14	Consistent Use of Labels	(VQC8)	corrected
Overlap in Disjoint Label Properties	S13	Consistent Use of Labels	VQC12, (VQC8)	corrected
Consistent Use of Mapping Properties	S46	Consistent Usage of Mapping Properties	(VQC8)	not checked
Disjoint Semantic Relations	S27	Consistent Usage of Semantic Relations	(VQC8)	corrected
Cycles in `broader` Hierarchy	-	-	VQC3	corrected
Extra Whitespace	-	-	-	corrected

conditions. We tried to keep our list of criteria as granular as possible. However, the SKOS integrity conditions S9 and S37 both define related class disjointness axioms, so we chose not to separate between them.

Based on an analysis of the potential severity of quality issues and some previous experience of problems with vocabularies published on the ONKI vocabulary service [21], we settled on eleven criteria, described further below. Nine of these criteria are taken from the PoolParty checker (with the PoolParty Consistent Use of Labels check split into two distinct criteria). Of the remaining criteria, one appears only in the qSKOS list of criteria (Cycles in `broader` Hierarchy) while one criterion, Extra Whitespace, is our own addition which we have found to cause problems in vocabularies published using ONKI.

Valid URIs The validity of resource URIs can be considered on many levels, from syntactical (the use of valid characters) to semantic (e.g. registered URI schemes) and functional (e.g. dereferenceability of HTTP URIs). The PoolParty checker only appears to perform a syntactic check. The qSKOS criterion VQC5 *Link Target Availability* is a stricter criterion, which requires that links are dereferenceable and the targets reachable on the Web.

Missing Language Tags RDF literals used for, e.g., concept labels may or may not have a specified language tag. Missing language tags are problematic, especially for multilingual vocabularies. The PoolParty checker counts the number of concepts, which have associated literal values without a language tag. The qSKOS criterion VQC11a *Language tag support* addresses the same issue.

Missing Labels `Concept`s and `ConceptScheme`s in the vocabulary should carry human-readable labels such as `prefLabel` or `rdfs:label`. The PoolParty checker verifies that this is the case. The qSKOS criterion VQC10 *Human readability* addresses the same issue, though the set of SKOS constructs and label properties to check is longer than in the PoolParty checker.

Loose Concepts The PoolParty checker defines loose concepts as `Concept` instances that are not *top concepts* (i.e. having incoming `hasTopConcept` or outgoing `topConceptOf` relationships) in any `ConceptScheme` and have no `broader` relationships pointing to other concepts. The checker counts the number of such loose concepts. The qSKOS quality criterion VQC1 *Relative number of loose concepts* is similarly named, but is a stricter criterion: qSKOS defines loose concepts as those concepts that don't link to any other concepts using SKOS properties.

Disjoint OWL Classes The SKOS specification defines that all the classes `Concept`, `Collection` and `ConceptScheme` are pairwise disjoint, that is, no resource may be an instance of more than one of these classes [14]. The PoolParty checker verifies that this is the case. qSKOS does not have an explicit criterion for this issue, but it is implicitly covered by VQC8 *SKOS semantic consistence*, which addresses all the SKOS consistency criteria.

Ambiguous `prefLabel` values The SKOS specification defines that "[a] resource has no more than one value of `prefLabel` per language tag" [14]. The PoolParty checker verifies this as a part of the Consistent Use of Labels check. This issue is also implicitly covered by VQC8 in qSKOS.

Overlap in Disjoint Label Properties The SKOS specification defines that the label properties `prefLabel`, `altLabel` and `hiddenLabel` are pairwise disjoint, i.e. a concept may not have the same label in more than one of these properties [14]. This is also verified as a part of the Consistent Use of Labels check in the PoolParty checker. The qSKOS criterion VQC12 *Ambiguous labeling* addresses the same issue, but it is also implicitly covered by VQC8.

Consistent Use of Mapping Properties The SKOS specification defines that the `exactMatch` relation is disjoint with both `broadMatch` and `relatedMatch`; that is, two `Concept`s cannot be mapped to each other using both `exactMatch` and one of the other mapping properties. The PoolParty checker verifies this in the Consistent Usage of Mapping Properties check. qSKOS does not have a specific criterion for this issue, but it is implicitly covered by VQC8.

Disjoint Semantic Relations The SKOS specification defines the `related` relation to be disjoint with `broaderTransitive`; that is, two concepts cannot be connected by both [14]. The PoolParty checker verifies this in the Consistent Usage of Semantic Relations check. This issue is also implicitly covered by the qSKOS criterion VQC8.

Cycles in `broader` Hierarchy Cycles in the hierarchies of terminological vocabularies can be simple mistakes that can arise when a complex vocabulary is created, but in some cases the cycles may carry important meaning [17]. Cycles are not forbidden by the SKOS specification and the PoolParty checker does not check for them. However, cycles can cause problems for automated processing such as term expansion in information retrieval systems, where any concept participating in a cycle may be considered equivalent to all the other concepts in the cycle. This issue is addressed by the qSKOS criterion VQC3 *Cyclic Relations*.

Extra Whitespace SKOS vocabularies may contain surrounding whitespace in label property values such as `prefLabel`, `altLabel` and `hiddenLabel`. While extra whitespace is not forbidden by SKOS, it is unlikely to carry meaning and may cause problems when the vocabulary is, e.g., stored in a database or used for information retrieval, particularly when exact string matching is performed. Such extra whitespace is likely an artifact of conversion from another textual format such as XML or CSV, or it may originate in the text fields of graphical user interfaces used for vocabulary editing, where whitespace is typically invisible.

Table 2. Vocabularies used in validation tests, grouped by size (small, medium and large). When version is not indicated, the latest available SKOS file on 9th January 2012 was used. The columns Conc, Coll and CS show number of concepts, collections and concept schemes, respectively.

Name	Version	Publisher	Description	Conc	Coll	CS
STI Subjects	-	NASA	Subject classification of spacefaring terms	88	0	0
NYT Subjects	-	New York Times	Subject descriptors used in NY Times data	498	0	0
GBA Thesaurus	-	Geological Survey Austria	Thesaurus of geological terms	780	0	2
NYT Locations	-	New York Times	Geographical locations used in NY Times data	1920	0	0
IAU Thesaurus 1993 (IAUT93)	-	IVOA	Legacy astronomical thesaurus	2551	0	1
IVOA Thesaurus (IVOAT)	-	IVOA	Astronomical thesaurus	2890	0	1
GEMET	3.0	EIONET	Environmental thesaurus	5208	79	1
STW Thesaurus	8.08	ZBW	Economics thesaurus	6621	0	12
Schools Online Thesaurus (ScOT)	-	Education Services Australia	Terms used in Australian and New Zealand schools	8110	0	1
Medical Subject Headings (MeSH)	2006 [4]	US NLM	Biomedical vocabulary	23514	0	0
Finnish General Thesaurus (YSA)	2012-01-09	National Library of Finland	General thesaurus used in Finnish library catalogs	24206	61	1
SWD subject headings	07/2011	DNB	Subject headings used in German library catalogs	166414	0	0
LCSH	2011-08-11	Library of Congress	Subject headings used in Library of Congress catalog	407908	0	18
DBpedia Categories	3.7	DBpedia project	Categories from Wikipedia	740362	0	0

4 Validity of SKOS Vocabularies

To gain an understanding of the current quality of SKOS vocabularies published online, we first collected 14 freely-available vocabularies in SKOS format, published by 12 different organizations. Most of the vocabularies were discovered from the SKOS wiki[5]. Among the vocabularies that were available as file downloads, we selected vocabularies based on three criteria: 1) geographical diversity, 2) topical diversity and 3) diversity of vocabulary sizes. In two cases (NY Times and IVOA) we chose two vocabularies per publisher in order to compare vocabulary quality within the same publisher.

The vocabularies, together with some general statistics about the number of concepts, SKOS collections, concept schemes and RDF triples, are shown in Table 2. The vocabularies can be roughly categorized by size: *small* (fewer than 2000 concepts), *medium* (between 2000 and 10000 concepts) and *large* (more than 10000 concepts). We then analyzed most of these vocabularies using the PoolParty checker.

Table 3. Validation and Correction Results. The first group of columns shows the result of validating the vocabularies using the PoolParty checker. Of these, the last four columns represent mandatory checks that must be passed for the vocabulary to be considered valid by the PoolParty checker. The second group of columns shows the number of problems in each vocabulary that were found and corrected by Skosify.

	Valid URIs	Missing Language Tags	Missing Labels	Loose Concepts	Disjoint OWL Classes	Consistent Use of Labels	Consistent Use of Mapping Properties	Consistent Use of Semantic Relations	Missing Language Tags	Missing Labels	Loose Concepts	Disjoint OWL Classes	Ambiguous `prefLabel` values	Overlap in Disjoint Label Properties	Disjoint Semantic Relations	Cycles in `broader` Hierarchy	Extra Whitespace
STI Subj.	pass	88	pass	1	pass	pass	pass	pass	3134	0	1	0	0	0	0	0	88
NYT Subj.	pass	0	pass	498	pass	pass	pass	pass	0	1	498	0	0	0	0	0	2
GBA	pass	0	pass	0	pass	pass	pass	pass	0	0	1	0	0	0	0	0	30
NYT Loc.	pass	0	pass	1920	pass	**fail**	pass	pass	0	1	1920	0	0	0	0	0	0
IAUT93	pass	358	**fail**	1060	pass	**fail**	pass	**fail**	358	1	1060	0	0	1	10	0	40
IVOAT	pass	2890	pass	926	pass	pass	pass	**fail**	7330	1	926	0	0	0	11	6	0
GEMET	pass	3	**fail**	109	pass	pass	pass	**fail**	3	0	109	0	0	0	2	0	0
STW	pass	2	**fail**	0	pass	pass	pass	**fail**	2	0	0	0	0	0	7	0	2
ScOT	pass	0	pass	0	pass	**fail**	pass	**fail**	0	0	0	0	0	1	26	0	1
MeSH	pass	0	pass	189	pass	pass	pass	**fail**	0	0	189	0	0	0	383	12	22610
YSA	pass	0	**fail**	8614	**fail**	pass	pass	**fail**	0	0	8614	61	0	0	58	6	0
SWD									0	0	65363	0	2	127	108	2	42
LCSH									0	0	423010	0	0	18	200	0	0
DBpedia									0	0	90822	0	0	0	10100	6168	0

4.1 Validation Results

For the first eleven vocabularies, we used the PoolParty checker to look for problems. The results of these checks are summarized in Table 3. Some vocabularies had to be pre-processed before validation[6]. The three largest vocabularies – DNB SWD, LCSH and DBpedia Categories – were too large for the checker.

Only the GBA Thesaurus passed all checks without problems. All the medium-size and large vocabularies that we tested failed at least one mandatory check, meaning that they did not meet some of the SKOS integrity constraints.

[5] `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`

[6] The GBA Thesaurus used an invalid character encoding, which we corrected. The IVOA Thesaurus used a deprecated SKOS namespace which we corrected. GEMET and MeSH consisted of several RDF files, which we merged into a single file. We converted MeSH into Turtle syntax and removed some vocabulary-specific attributes to keep it below the 20MB limit of the PoolParty checker.

5 Correcting Problems

To further analyze structural problems in SKOS vocabularies and to correct as many of them as possible, we created the **Skosify** tool. It is a command line utility that reads one or more SKOS files as input, performs validation checks and structural corrections on the vocabulary, and outputs a corrected SKOS file. Skosify was implemented in Python using the rdflib[7] toolkit. It has been released[8] as open source under the MIT License. An online version of the tool is also available[9].

We processed all the vocabularies listed in Table 2 with Skosify[10]. The number of corrections performed for each vocabulary is shown in the last group of columns of Table 3.

5.1 Addressing the Validity Criteria

Of the eleven validation criteria defined in Section 3, Skosify as of version 0.5 addresses nine, as shown in the last column of Table 1. The criteria for Valid URIs and Consistent Use of Mapping Properties are not addressed in the current version; however, none of the vocabularies we tested violated these according to the PoolParty checker. Corrections performed by Skosify for the remaining criteria are described in the following subsections.

Missing Language Tags The STI Subjects and both IVOA astronomical thesauri have a large number[11] of concepts with label property values lacking language tags. GEMET and the STW Thesaurus also lack a few language tags. Skosify was able to correct these when given a *default language* setting. However, this approach only works when the language of untagged literals is known and different languages have not been mixed.

Missing Labels Four vocabularies have either concept schemes or concepts without labels. Skosify can detect unlabeled concept schemes and optionally, when given a label as parameter, add the missing label. However, Skosify does not detect unlabeled concepts.

[7] `http://rdflib.net`

[8] `http://code.google.com/p/skosify/`

[9] `http://demo.seco.tkk.fi/skosify`

[10] The SVN repository of Skosify includes a test suite for processing all the above mentioned vocabularies as well as PoolParty checker reports before and after Skosify processing. The processed vocabularies have been provided as a separate download.

[11] The reported number of missing language tags in Table 3 is sometimes different for the different tools, because the PoolParty checker groups the missing language tags by concept, while Skosify counts every label without language tag separately.

Loose Concepts Most of the vocabularies we examined contain loose concepts. The STI Subjects, both NY Times vocabularies, MeSH, SWD and DBpedia Categories do not include any `ConceptScheme` instances, so the existence of loose concepts was a natural consequence. GEMET, YSA and LCSH[12] do include one or more `ConceptScheme` instances, but do not use any `hasTopConcept` or `topConceptOf` relationships to define top concepts, again leading to loose concepts. Both the IVOA vocabularies use `topConceptOf` relationships, but do not mark all of the top level concepts using those properties. In these cases, Skosify identifies the top level concepts (those with no `broader` relationships) and adds `hasTopConcept` and `topConceptOf` relationships to a concept scheme, creating one if necessary.

Disjoint OWL Classes YSA was the only vocabulary that failed the Disjoint OWL Classes test in the PoolParty checker. In this case, the problem was that some relationships intended for `Concept`s, such as `exactMatch`, were used on `Collection` instances. The RDFS inference capabilities of the PoolParty checker together with `rdfs:domain` specifications of some SKOS properties caused those instances to be marked both as `Concept`s and `Collection`s. Skosify identifies this particular error and corrects it by removing the improper relationship assertions. However, it cannot correct the more general case where a resource is explicitly marked as being of several types that are defined to be disjoint.

Ambiguous `prefLabel` values Of the vocabularies we tested, only the SWD subject headings was found by Skosify to contain concepts with more than one `prefLabel` using the same language tag. On a closer look, the two cases appeared to be genuine errors: in one case, the term *Einheitshaus* is used in the same concept with *TURBO C++ für WINDOWS*, and in the other case, *Hämatogene Oxidationstherapie* appears together with *Pikkoloflötenspiel*. In these situations, Skosify arbitrarily selects one of the labels as the real `prefLabel` while the rest are converted into `altLabel`s. By default, Skosify will choose the shortest `prefLabel`, but other options are available for choosing the longest label or not performing any correction at all.

Overlap in Disjoint Label Properties Four of the vocabularies we tested contain cases where a concept is linked to a label using two different label properties that are defined as disjoint by the SKOS specification. An example from ScOT is shown in Figure 1a. In this situation, Skosify removes the value for the less important property (`hiddenLabel` < `altLabel` < `prefLabel`).

[12] In LCSH, every concept exists in two concept schemes. The number of loose concepts for LCSH in Table 3 is higher than the total number of concepts in LCSH because loose concepts are determined per concept scheme, so the same concept may be counted twice. In total, LCSH has 211505 different loose concepts.

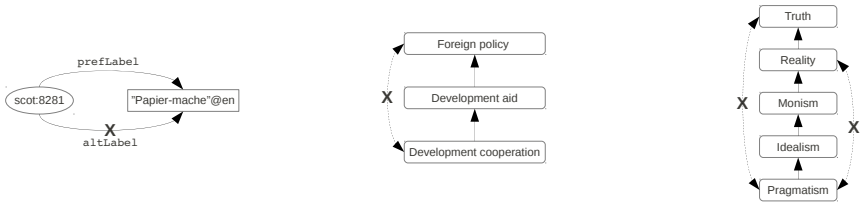

Fig. 1. Examples of overlapping labels (a) and disjoint semantic relations (b and c). In subfigures b and c, line arrows represent `broader` relationships while dotted arrows represent `related` relationships between concepts. Crosses (X) mark relationships that were eliminated by Skosify.

Disjoint Semantic Relations Ten of the vocabularies we tested (all but the four smallest) contain cases where a concept is linked to another concept using relationships that are defined as disjoint by the SKOS specification. In particular, the `related` relationship is often used to link between concepts that are directly above or below each other in the `broader` hierarchy, as shown in Figure 1b and 1c. In this situation, Skosify removes the `related` relationship assertion, leaving the `broader` hierarchy intact. This correction is performed by default, in order to enforce the SKOS integrity condition S27, but it can be disabled.

Cycles in `broader` Hierarchy In the vocabularies we examined, we found many examples of cycles in the `broader` hierarchy. Some examples of these are shown in Figure 2.

The simplest kind of cycle is a concept which has a `broader` relationship to itself, as in Figure 2a. Another simple case is when a concept has a `broader` relationship to its child, as shown in Figure 2b. In both these cases the relationship causing the cycle is probably an error and can be rather safely eliminated.

More complex cases are cycles where the concepts participating in the cycle are both on the same hierarchy level (i.e. have the same minimum path length to top-level concepts), as in Figures 2c and 2d. In these cases it is difficult to automatically select the relationship to be eliminated. It is still likely that the cycle is a mistake, but properly correcting it may require human intervention.

MeSH 2006 contains several cycles in the SKOS version. They arise because MeSH is structured along several hierarchies (facets) which each have their own criteria for hierarchical relationships. The SKOS version aggregates these hierarchies so that the distinctions between facets are lost. One interesting cycle is the one involving the concepts Morals and Ethics, shown in Figure 2e. This cycle appears to be intentional[13] and it still exists in MeSH 2012.

DBpedia Categories contain thousands of cycles. The DBpedia authors note that the "categories do not form a proper topical hierarchy, as there are cycles in

[13] For some discussion on the issue, see: `http://lists.w3.org/Archives/Public/public-esw-thes/2011Jul/0005.html`.

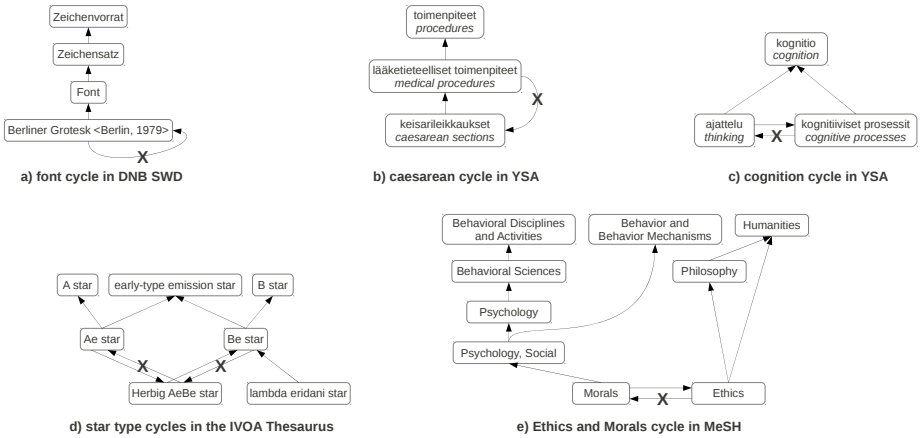

Fig. 2. Examples of cycles from the test vocabularies. English equivalents for Finnish terms are shown in *italic*. Arrows represent `broader` relationships between concepts. Crosses (X) mark relationships that were eliminated by Skosify during a particular run.

the category system and as categories often only represent a rather loose relatedness between articles" [6]. The cycles arise because the semantics of Wikipedia categories do not match traditional terminological hierarchies.

Skosify can detect and optionally remove cycles in the `broader` hierarchy. It uses a naïve approach based on performing a depth-first search starting from the topmost concepts in the hierarchy. In Figure 2, the relationships removed by Skosify during a test run with cycle elimination enabled have been crossed out.

The depth-first search approach for eliminating cycles is simple, fast and domain independent, but may not produce deterministic results and "cannot ensure that the links ignored during the graph traversal in order to prevent loops from happening are actually the appropriate links to be removed" [15]. More accurate formal methods for eliminating cycles in terminological hierarchies exist, but they are more complex and not as general as the naïve approach [15].

Despite the limitations of the naïve cycle elimination approach, it can be useful for alerting vocabulary maintainers of possible problems. For example, the maintainers of YSA were alerted of the existence of cycles detected by Skosify, including those shown in Figures 2b and 2c, and these have since been eliminated both from the original vocabulary and its SKOS version.

Extra Whitespace Eight of the vocabularies we tested contain SKOS label or documentation property values with surrounding whitespace. MeSH is particularly prone to this, with over 22000 such cases. Skosify removes the extra whitespace from these values.

5.2 Other Transformations

SKOS includes some redundant ways of representing information. Hierarchical relationships can be expressed using either `broader` or `narrower` (or both). Likewise, top-level concepts in a concept scheme can be described using either `hasTopConcept` or `topConceptOf` [14]. Additionally, transitive properties such as `broaderTransitive` and general properties such as `semanticRelation` can in principle be inferred using RDFS/OWL inference. Whether such redundant information is desirable depends on the needs of the application: for example, the Helping Interdisciplinary Vocabulary Engineering tool[14] requires hierarchical relationships to be specified using both `broader` and `narrower`. The qSKOS quality criteria include a *Redundant Relationships* criterion which seeks to measure the amount of redundancy in a SKOS vocabulary.

Skosify has support for generating either a minimally redudant version of the hierarchy of a SKOS vocabulary (using only `broader` relationships), or a version with both `broader` and `narrower` relationships. It can also be used to explicitly assert transitive relationships.

Skosify can also be given arbitrary RDF files as input, together with a mapping configuration specifying how the RDF constructs used in the input correspond to SKOS constructs. We have used this capability to transform many lightweight OWL ontologies created in the FinnONTO projects [21] into structurally valid SKOS versions, which are published alongside the original OWL versions. This capability is described in more detail in the Skosify wiki[15].

We have also found that some FinnONTO vocabularies contain remnants of previously used RDF lists, consisting mostly of blank nodes. Skosify removes such RDF graph elements that are disconnected from the main vocabulary graph.

6 Evaluation

For evaluating how well we attained the goal of improving SKOS vocabularies, we considered 1) improvements in vocabulary validity resulting from its use; 2) the performance and scalability of Skosify when used with large vocabularies.

6.1 Improvements in Validity

We revalidated the SKOS vocabularies processed by Skosify using the PoolParty checker using the same methodology as described in Section 4, again excluding the three largest vocabularies. In most cases, the validation problems shown in Table 3 had indeed been corrected. However, some problems remained.

GEMET, STW Thesaurus and YSA still failed the Missing Labels check. GEMET contains concepts withous labels, and since Skosify did not attempt to correct this issue, the outcome was expected. For the STW Thesaurus and YSA, the problem was caused by a concept scheme being labeled with a `prefLabel`.

[14] `http://code.google.com/p/hive-mrc/`
[15] `http://code.google.com/p/skosify/wiki/GettingStarted`

This property is not recognized by the PoolParty Checker, which only looks for `rdfs:label` properties of concept schemes.

The NY Times Locations vocabulary still did not pass the Consistent Use of Labels check. The vocabulary contains different descriptions of geographical locations, interlinked using `owl:sameAs` relationships. Skosify does not perform OWL inference, so it did not identify cases where the same location was named using different `prefLabels` for different resource URIs. The PoolParty checker performs `owl:sameAs` inference so it was able to detect these inconsistent labels.

Skosify found one loose concept in the GBA Thesaurus, despite it having passed the PoolParty check for loose concepts. This discrepancy is caused by the GBA Thesaurus not using any explicit `inScheme` relationships despite containing two `ConceptScheme` instances. Skosify is unable to infer which concept scheme(s) concepts belong to and thus it misidentifies one concept (*Lithstrat*) as being loose, even though it is marked as a top concept of one of the concept schemes.

6.2 Performance

On a modern desktop PC (Intel Core i5 CPU, 8GB RAM), processing of the largest vocabularies, LCSH and DBpedia Categories, took 25 minutes and 90 minutes, respectively. Memory usage was below 4GB in each case. Many vocabularies published on the ONKI ontology service [20] are automatically processed with Skosify as a part of the publication process.

7 Discussion

In this study, we first looked at the possible criteria for determining the validity and quality of SKOS vocabularies. We created a list of eleven criteria that we collected and synthetized from several sources.

We then surveyed freely available SKOS vocabularies for different domains and measured their validity, mainly using the PoolParty checker. We found that most vocabularies, particularly the larger ones, violate one or more of the SKOS integrity conditions. This is not surprising, since RDF data published online has been found to contain many errors [9,10,11]. However, earlier studies did not look specifically at the validity of SKOS vocabularies. In particular, we found that the SKOS integrity condition S27, which specifies that the `related` relationship is disjoint with the `broaderTransitive` relationship, is violated by nearly all of the vocabularies we examined. The YSA maintainers specifically declined to remove some relationships violating this constraint from the vocabulary, because they considered them important for vocabulary users. A similar argument could be made for LCSH, which has a very complex `broader` hierarchy with a large amount of multiple inheritance. Thus, the constraint in its current form could be considered overly strict. It could be amended by only forbidding `related` relationships between direct descendants rather than considering the whole transitive hierarchy.

Finally, we created the Skosify tool and used it to improve the validity of fourteen SKOS vocabularies. Our tool was able to correct the great majority of identified structural problems in these vocabularies. If the same processing and validation were performed on a larger selection of SKOS vocabularies, we expect new kinds of problems to be found. Still, the corrections performed by Skosify appear to be useful for many different vocabularies. The implementation is fast enough to be used routinely as a part of the publication process for SKOS vocabularies. Skosify can also be used as a validation tool, particularily for large vocabularies which can be difficult to process using online tools.

In future work, we intend to examine a wider selection of vocabularies and to evaluate the correction results against other validation tools such as the qSKOS tool. The online version of Skosify could be further developed to expose more of the functionalities of the command line version and to better support validation.

Acknowledgments

This work is part of the National Semantic Web Ontology project in Finland FinnONTO[16] (2003-2012), funded mainly by the National Technology and Innovation Agency (Tekes) and a consortium of 38 public organizations and companies. We thank Christian Mader for assistance in relating the qSKOS criteria to other sources, Jouni Tuominen, Matias Frosterus and Eeva Kärki for correcting structural problems in YSA, and Andreas Blumauer and Alexander Kreiser for technical assistance with the PoolParty checker as well as for providing this excellent and free online validation service in the first place.

References

1. ISO 2788: Guidelines for the establishment and development of monolingual thesauri. International Organization for Standardization (ISO) (1986)
2. Structured vocabularies for information retrieval. Part 2: Thesauri (BS 8723-2:2005). British Standards Institution (2005)
3. Aitchison, J., Gilchrist, A., Bawden, D.: Thesaurus Construction and Use: A Practical Manual. Aslib IMI (2000)
4. van Assem, M., Malaisé, V., Miles, A., Schreiber, G.: A method to convert thesauri to SKOS. In: Sure, Y., Domingue, J. (eds.) The Semantic Web: Research and Applications, Lecture Notes in Computer Science, vol. 4011, chap. 10, pp. 95–109. Springer, Berlin, Heidelberg (2006)
5. Berrueta, D., Fernández, S., Frade, I.: Cooking HTTP content negotiation with Vapour. In: Proceedings of 4th workshop on Scripting for the Semantic Web (SFSW2008) (2008)
6. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154 – 165 (2009)
7. Borst, T., Fingerle, B., Neubert, J., Seiler, A.: How do libraries find their way onto the Semantic Web? Liber Quarterly 19(3/4) (2010)

[16] `http://www.seco.tkk.fi/projects/finnonto/`

8. Byrne, G., Goddard, L.: The strongest link: Libraries and linked data. D-Lib Magazine 16(11/12) (2010)
9. Ding, L., Finin, T.: Characterizing the semantic web on the web. Electrical Engineering 4273(August), 5–9 (2006)
10. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the pedantic web. In: Proceedings of the 3rd International Workshop on Linked Data on the Web (LDOW2010) (2010)
11. Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of Linked Data conformance. Web Semantics: Science, Services and Agents on the World Wide Web 14, 14–44 (July 2012)
12. Mader, C., Haslhofer, B.: Quality criteria for controlled web vocabularies. In: Proceedings of the 10th Europen Networked Knowledge Organisation Systems Workshop (NKOS 2011) (2011)
13. Malmsten, M.: Making a library catalogue part of the semantic web. In: Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications. pp. 146–152. Dublin Core Metadata Initiative (2008)
14. Miles, A., Bechhofer, S.: SKOS simple knowledge organization system reference. World Wide Web Consortium Recommendation (August 2009), http://www.w3.org/TR/skos-reference/
15. Mougin, F., Bodenreider, O.: Approaches to eliminating cycles in the UMLS Metathesaurus: Naïve vs. formal. In: American Medical Informatics Association (AMIA) Annual Symposium Proceedings. pp. 550–554 (2005)
16. Nagy, H., Pellegrini, T., Mader, C.: Exploring structural differences in thesauri for SKOS-based applications. In: I-SEMANTICS 2011. Graz, Austria (September 2011)
17. Nebel, B.: Terminological cycles: Semantics and computational properties. In: Principles of Semantic Networks. Morgan Kaufmann, San Mateo, CA (1991)
18. Schandl, T., Blumauer, A.: PoolParty: SKOS thesaurus management utilizing linked data. In: Proceedings of the 7th Extended Semantic Web Conference (ESWC2010) (2010)
19. Summers, E., Isaac, A., Redding, C., Krech, D.: LCSH, SKOS and Linked Data. In: Proceedings of the International Conference on Dublin Core and Metadata Applications (DC-2008). pp. 25–33. Dublin Core Metadata Initiative (Sep 2008)
20. Tuominen, J., Frosterus, M., Viljanen, K., Hyvönen, E.: ONKI SKOS server for publishing and utilizing SKOS vocabularies and ontologies as services. In: Proceedings of the 6th European Semantic Web Conference (ESWC 2009). Springer-Verlag (May 31 - June 4 2009)
21. Viljanen, K., Tuominen, J., Hyvönen, E.: Ontology libraries for production use: The Finnish ontology library service ONKI. In: Proceedings of the 6th European Semantic Web Conference (ESWC 2009). Springer-Verlag (May 31 - June 4 2009)