Streamlining Vocabulary Conversion to SKOS: A YAML-based Approach to Facilitate Participation in the Semantic Web

Christoph Göpfert[1[0000-0001-6659-8947]] Jan Ingo Haas[1[0000-0003-1112-3893]], Lucas Schröder[1[0009-0009-5540-4495]] and Martin Gaedke[1[0000-0002-6729-2912]]

[1] Technische Universität Chemnitz, 09111 Chemnitz, Germany
{christoph.goepfert,jan-ingo.haas,
lucas.schroeder,martin.gaedke}@informatik.tu-chemnitz.de

Abstract. Controlled vocabularies, such as classification schemes, glossaries, taxonomies, or thesauri, play an important role in many Web services. One of the main areas of application of controlled vocabularies is the domain of information retrieval systems, as they can be used to improve the findability of resources. For instance, concepts described in a vocabulary may be used to uniquely classify resources, to tag them with relevant keywords, or to annotate them with domain-specific attributes. The Simple Knowledge Organization System (SKOS) is an established data model of the Semantic Web domain that can be used to describe vocabularies in a semantically structured format. However, modelling a vocabulary is oftentimes highly time demanding, labor-intensive, and requires both familiarity with basic Semantic Web technologies and expertise in the application domain. This complicates both the development of new vocabularies and the conversion of existing vocabularies into the RDF data model. We propose an intermediate, YAML-based format to express concepts and their relationships hierarchically. The intermediate format can be converted automatically into a SKOS vocabulary using a command-line conversion program. To demonstrate the feasibility of our approach, we selected 26 vocabularies of highly diverse formats, expressed them in the proposed intermediate format, which was subsequently converted in an automated manner into the corresponding SKOS vocabulary using our yaml2skos program. Our approach enables users with little to no familiarity with the Semantic Web to develop SKOS vocabularies, thereby lowering the barrier to participation in the Semantic Web landscape.

Keywords: Vocabulary, Controlled Vocabulary, Conversion, Simple Knowledge Organization System, SKOS, Semantic Web, Linked Data, Taxonomy, Glossary, Thesaurus.

1 Introduction

Controlled vocabularies are a widely used way of representing knowledge such as glossaries, taxonomies, or thesauri. The application areas of controlled vocabularies are extremely diverse, ranging from the classification of cultural works [1], the classification

of literature or publications [2, 3] to various thesauri [4, 5] and taxonomies [6]. A major area of application of controlled vocabularies can be found in information retrieval, especially in systems utilizing keyword-based search methods. Performing simple syntactic comparisons between search terms and keywords used for annotation of resources often leads to incorrect search results [7, 8]. Through the use of controlled vocabularies, these can be improved [8].

The Simple Knowledge Organization System[1] (SKOS) offers a semantically structured data model for representing controlled vocabularies. In the literature, the use of the terms "glossary", "taxonomy", "thesaurus", "vocabulary" and "ontology" is often ambiguous. Our proposed approach can be used with any knowledge model mappable to the SKOS data model. For this reason, we use the term vocabulary as a collective term for the aforementioned terms in the subsequent sections.

The development of vocabularies in SKOS requires an understanding of Semantic Web technologies, especially the Resource Description Framework[2] (RDF) data model and RDF formats. In addition, the effort required to develop or to translate a vocabulary into SKOS can be time demanding.

In this paper, we present an approach for creating SKOS vocabularies using an intermediate format. Using this intermediate format, new vocabularies can be created from scratch and existing vocabularies that can be mapped to SKOS can be expressed as well. The intermediate format's structure is intended to be as human-readable as possible to facilitate vocabulary creation. In addition, the verbosity compared to common RDF formats is reduced, which in turn cuts down on the amount of typing required.

The rest of the paper is structured as follows. In section **Fehler! Verweisquelle konnte nicht gefunden werden.**, related work is reviewed. Section 3 presents the structure of the proposed intermediate format and details the conversion of the intermediate format to SKOS. Section 4 evaluates the proposed approach by expressing 26 taxonomies of diverse formats in the intermediate format, converting them to SKOS and assessing the quality of the resulting SKOS vocabularies. Finally, section 5 outlines conclusions.

2 Related Work

van Assem et al. [9] introduce an approach for converting structured data (like WordNet[3]) onto RDF. The proposed idea features mapping the syntactical elements as well as the structural hierarchy of the source format onto RDF using OWL as a meta format, augmenting the resulting metamodel with semantic properties that did not exist in the original model. This step allows for interpretation of the source model. As a last step, the metamodel is standardized, i.e. transferred into SKOS. Our approach differs in two fundamental ways. Firstly, the described approach requires the data to be mapped to be available in a structured format. Our approach allows for conversion of any format to SKOS, as long as it can be represented in SKOS. Secondly, we forgo the syntactical

[1] https://www.w3.org/TR/skos-reference/
[2] https://www.w3.org/TR/rdf11-concepts/
[3] https://wordnet.princeton.edu/

mapping of the source data onto a metamodel. Instead we use SKOS as our destination model, directly converting to it via an intermediate, semantically equivalent format.

The authors of [10] make a distinction between “thesauri with standard structure” and “thesauri with non-standard structure”. While the first can be easily transferred to SKOS, the latter has semantics that are not inherent in the SKOS standard and thus for the purpose of integration into SKOS, an extension to declare subclasses has to be developed. An adaption of this idea is proposed in [11]. Common to both approaches is the need for the implementation of a program that performs the final mapping from the intermediate RDF format for each respective process. Our approach differs in this regard our intermediate format is built upon YAML, thus making the conversion into SKOS fully automated for every conversion process.

More recent approaches aim to semi-automate the conversion process. A notable example of this is Skosify [12] which accepts RDF graphs and with the help of a mapping scheme transforms those into SKOS. By specifying the configuration file for the mapping scheme, the previously mentioned approaches are effectively formalized. However, the conversion is not fully automated as it requires the data to exist in an RDF format. Our approach avoids the use of existing RDF formats to perform conversions. Instead we chose to extend YAML as an easy and human-readable format.

YAML is a popular choice for an intermediate format, notable examples are yaml2sbml [13], a tool to convert from a YAML based language to SBML and YARRRML [14] which can be employed when converting from various data sources to an RDF format using a simplified and human readable representation of RML rules.

Another approach to the development of vocabularies are programs that offer WYSIWYG interfaces. A popular representative of this approach is the tool Protegé [15]. Protegé is a software framework for creating and editing ontologies.

3 Vocabulary Conversion

We introduce a novel third approach that does not assume the user to have any prior knowledge of RDF formats. Instead, we introduce an intermediate format for describing vocabularies. This intermediate format can then be processed using the *yaml2skos* conversion program we developed to generate an equivalent SKOS vocabulary in a desired RDF format. The structure of the intermediate format and the conversion to the SKOS vocabulary are detailed in the following sections.

3.1 Intermediate Format Design

Developing a vocabulary in an RDF format requires the developer to have some knowledge of the RDF data model and its formats, which is an obvious obstacle to participation in the Semantic Web. Moreover, the primary motivation behind the RDF data model was to design a format that is easily machine-processable. Common RDF formats do by default not visually represent the hierarchical structure of concepts well, which complicates spotting errors in the hierarchy of a vocabulary for users.

The SKOS data model is characterized by its simplicity, as reflected in the relatively small number of classes and properties. With the intermediate format presented below,

we are pursuing the objective of maintaining this simplicity by reflecting it in the format, i.e. in its user-friendly notation and reduction of superfluous statements.

The intermediate format we propose bases on the YAML format and extends on terms defined by the SKOS Core vocabulary. YAML has been used before as a format for intermediate files in similar scenarios [13, 16], due to its human-friendly structure. As the intermediate file is to be used by our *yaml2skos* program to automatically generate the equivalent SKOS vocabulary, its expressiveness must be sufficient to represent instances of all classes and properties defined in the SKOS core vocabulary.

```yaml
meta:
    id: hri
    uri: https://purl.org/net/vsr/taxannot/hri
    dcterms:
        - title: A Taxonomy to Structure and Analyze Human-Robot Interaction
        - creator: Linda Onnasch
        - creator: Eileen Roesler
```

Listing 1: "meta" section including DCMI Terms metadata (in YAML)

```turtle
@prefix dct: <http://purl.org/dc/terms/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<https://purl.org/net/vsr/taxannot/hri#hri> a skos:ConceptScheme ;
    dct:creator "Eileen Roesler"@en,
        "Linda Onnasch"@en ;
    dct:title "A Taxonomy to Structure and Analyze Human-Robot Interaction"@en ;
    skos:hasTopConcept <https://purl.org/net/vsr/taxannot/hri#fields-of-application> .
```

Listing 2: SKOS excerpt generated from "meta" section (in Turtle format)

The structure of the intermediate format is separated into two main sections, a meta section, and a remaining section for the description of the actual concepts, optionally as part of ordered or unordered collections that may be included in the vocabulary. A complete list of the available terms of the intermediate format can be found in our openly accessible repository[4]. The meta section is intended to specify general meta information about the vocabulary. It is required to specify a default namespace which will subsequently be used for any concepts described in the vocabulary which no other namespace is explicitly stated for. Optionally, further namespaces may be specified as well. It is also required to specify an id for the vocabulary. Besides this information, further optional metadata may be specified using any terms of the DCMI Metadata Terms vocabulary[5], as shown in Listing 1. Any metadata provided in the meta section will be used to describe the vocabulary's *ConceptScheme* node that will be generated by the program automatically. An example for this is shown in Listing 2, which *yaml2skos* generated using the data shown in Listing 1 as input.

The remaining section is intended for the definition of concepts, optionally as part of an ordered or unordered collection. The intermediate format does not require collections to be defined, however, it requires at least one concept to be defined. In order to define a concept to be an upper concept of another concept, users may use a nested

[4] https://purl.org/net/vsr/taxannot/yaml2skos
[5] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

notation style to do so. An example of this notation is shown in Listing 3 (left). By using an indented notation style, the hierarchical relation of the concepts is visualized for the user, which makes it easier to locate potential errors in the hierarchy.

```yaml
concepts:
  fields-of-application:
    - l: Fields of Application
    - narrower:
      - industry:
        - l: Industry
        - def: Industry is ...
      - service:
        - l: Service
        - def: Service is ...
```

```yaml
concepts:
  fields-of-application:
    - l: Fields of Application
  industry:
    - l: Industry
    - def: Industry is ...
    - broader: fields-of-application
  service:
    - l: Service
    - def: Service is ...
    - broader: fields-of-application
```

Listing 3:" concepts" section using nested notation (left) and reference notation style (right)

```yaml
collections:
    collection_fields:
        - l: A first collection
        - industry
        - service
    collection_roles:
        - l: A second collection
        - supervisor
        - collaborator
```

Listing 4: Assigning concepts into collections

An alternative approach to model such relations can be achieved by referencing the corresponding concepts instead. This approach should be preferred for larger vocabularies, especially for vocabularies with deep hierarchies, as the nested notation would result in large indentations, negatively impacting readability. Listing 3 (right) shows an equivalent example, in which referencing is used instead of the nested notation.

The described concepts may be assigned to a collection. An example is shown in Listing 4, in which two collections are defined. A "collections"- or "ordered-collections"-section must be created. Then, collections and associated concepts can be listed.

3.2 Conversion to SKOS

Once a vocabulary has been expressed in the intermediate format, it can be converted with the *yaml2skos* program into an equivalent SKOS representation. The program processes each of the key-value pairs and generates corresponding triple statements. Users may specify a desired output format; most common RDF formats are supported.

The program validates the vocabulary using Skosify. Once the validation succeeds without severe issues, Skosify is used to enrich the generated vocabulary. Finally, the vocabulary will be output in SKOS. The program generates additional triples from implicit information and thus relieves the user of unnecessary tasks when modeling the vocabulary. This includes the automatic tagging of string literals to the desired language, or English by default. Concepts, collections, and the concept scheme node are

automatically assigned to their respective SKOS class. The vocabulary is described by the concept scheme node using information provided in the "meta" section.

4 Evaluation

To evaluate our approach, we selected 26 vocabularies, transferred them into the proposed intermediate format and converted them into SKOS using *yaml2skos*. Finally, we evaluated their quality using the qSKOS vocabulary assessment framework [17].

4.1 Vocabulary Selection

We conducted a systematic literature review following the procedure described by Kitchenham [18] to identify suitable controlled vocabularies. First, we set the scope to exclusively consider vocabularies related to the computer science domain as well as intersecting fields. We established two inclusion and two exclusion criteria to filter search results. Since our main objective was to identify vocabularies, we assumed that the title of a suitable search result suggests containing a vocabulary, otherwise the search result was discarded. If the title suggested containing a vocabulary, the content of the search result was inspected for the presence of a vocabulary. In case the search result did not contain a vocabulary, it was discarded. Furthermore, the search was limited to only considering the first 100 search results per search query and results in the English language.

The following search engines and catalogues were used as sources: Google Scholar, ACM Digital Library, IEEE Xplore, Springer Link, Science Direct, Web of Science and BARTOC. A total of 78 searches were carried out. After applying the in- and exclusion criteria, 57 search results remained. 18 results were removed as they were either duplicates, inaccessible or the described vocabulary was not expressible in SKOS. Further 13 vocabularies were excluded as they were either already available in SKOS or in OWL[6] (Web Ontology Language) format which also does not require a manual conversion as OWL can be converted using the tool Skosify. The formats of the remaining vocabularies were highly diverse, among others including HTML (5), images (12), tables (3), or text (2). The vocabularies converted to SKOS, including further vocabularies converted using Skosify, are available on Zenodo[7]. The 26 selected vocabularies are also listed on Zenodo[8], including their shortname in brackets which was used to refer to the respective vocabulary in the quality assessment in the next section 4.2.

4.2 Vocabulary Quality Assessment

For the 26 remaining vocabularies, representations in the intermediate formats were created, which were then used to generate equivalent SKOS vocabularies using the

6 https://www.w3.org/OWL/
7 https://doi.org/10.5281/zenodo.7908855
8 https://doi.org/10.5281/zenodo.10652215

yaml2skos program. We then analyzed the quality of the resulting SKOS vocabularies using quality metrics of the qSKOS vocabulary assessment tool [17]. The default configuration of qSKOS was used. The results of the quality assessment are available on Zenodo[8]. Quality metrics that were not satisfied by at least one or more vocabularies are addressed below:

Quality issues were indicated for metrics Q3 and Q13 due to some vocabularies containing concept clusters and orphaned concepts. These cases were no errors, but intended by design, as e.g. glossaries commonly contain "orphaned" concepts. Further issues were raised for metrics Q6, Q15 and Q24. In most cases, the cause could be found in the design of the source vocabulary. For instance, some concepts were explicitly related to multiple top-level concepts, violating Q6, or semantically similar concepts were sub-concept of different parent concepts, violating Q15. Further, explicitly stating relations in both directions in the intermediary format violated Q24, although this is not a modeling error per se, but superfluous information. As the evaluation using qSKOS examined only one vocabulary at a time, the tool was unable to recognize that some *skos:closeMatch* relationships related to concepts in external vocabularies. Consequently, the tool incorrectly assumed a relation via a mapping property to a member that is not part of a concept schema. This led to a poor score in metric Q10 which should be regarded a false positive. The poor ratings for metrics Q10 and Q21 were caused by their source vocabularies not containing labels or definitions for all concepts.

In summary, it can be observed that many of the encountered quality problems were already present in the source vocabularies. The quality of the generated vocabularies is therefore directly related to the quality of the source vocabularies.

5 Conclusion

In this paper, we propose an intermediate format for modeling vocabularies from which an equivalent SKOS vocabulary can be automatically generated. The intermediate format is intended to enable users with little or no familiarity with Semantic Web technologies to develop new or to transfer existing vocabularies into SKOS. The intermediate format is designed with a simple structure to provide users with an easily readable and understandable format. Vocabularies expressed in the introduced intermediate format can be automatically converted into a SKOS vocabulary using our *yaml2skos* tool.

To demonstrate the feasibility of our approach, we selected 26 vocabularies which were present in varying formats. These vocabularies were first transcribed into the YAML-based intermediate format before being automatically converted into the SKOS data model. The quality of the generated vocabularies was assessed using the qSKOS vocabulary quality assessment framework. The results of this assessment show that SKOS vocabularies can be successfully generated using our proposed approach.

Our approach makes an effort to reduce the barrier of developing SKOS vocabularies by providing a novel intermediate format that does not require familiarity with Semantic Web technologies to represent vocabularies.

Acknowledgement. . The research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID 514664767—TRR 386.

References

1. Introduction to Controlled Vocabularies: Terminologies for Art, Architecture, and Other Cultural Works.
2. The 2012 ACM Computing Classification System.
3. Library of Congress Classification Outline - Classification - Cataloging and Acquisitions (Library of Congress).
4. Scriven, M.: Evaluation Thesaurus. Edgepress (1981).
5. Thesaurus: mass communication - UNESCO Digital Library.
6. Gawron, V.J., Anno, G., Fleishman, E.A., Jones, E.D., Lovesey, E.J., McGlynn, L.E., McMillan, G., McNally, R.E., Meister, D., Brien, L.O., Promisel, D.M., Ramierez, T., Smith, B.L.: Human Factors Taxonomy. Proc. Hum. Factors Soc. Annu. Meet. 35, 1284–1287 (1991).
7. Gross, T., Taylor, A.: What Have We Got to Lose? The Effect of Controlled Vocabulary on Keyword Searching Results. Coll. Res. Libr. (2005).
8. Gross, T., Taylor, A.G., Joudrey, D.N.: Still a Lot to Lose: The Role of Controlled Vocabulary in Keyword Searching. Cat. Classif. Q. 53, 1–39 (2015).
9. van Assem, M., Menken, M.R., Schreiber, G., Wielemaker, J., Wielinga, B.: A Method for Converting Thesauri to RDF/OWL. In: The Semantic Web – ISWC 2004. pp. 17–31. Springer, Berlin, Heidelberg (2004).
10. SWAD-Europe Thesaurus Activity: Deliverable 8.8 Migrating Thesauri to the Semantic Web.
11. van Assem, M., Malaisé, V., Miles, A., Schreiber, G.: A Method to Convert Thesauri to SKOS. In: Sure, Y. and Domingue, J. (eds.) The Semantic Web: Research and Applications. pp. 95–109. Springer, Berlin, Heidelberg (2006).
12. Suominen, O., Hyvönen, E.: Improving the Quality of SKOS Vocabularies with Skosify. In: Knowledge Engineering and Knowledge Management. pp. 383–397. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33876-2_34.
13. Vanhoefer, J., Matos, M., Pathirana, D., Schälte, Y., Hasenauer, J.: yaml2sbml: Human-readable and -writable specification of ODE models and their conversion to SBML. J. Open Source Softw. 6, 3215 (2021).
14. Assche, D.V., Delva, T., Heyvaert, P., Meester, B.D., Dimou, A.: Towards a more human-friendly knowledge graph generation & publication.
15. Musen, M.A.: The Protégé Project: A Look Back and a Look Forward. AI Matters. 1, 4–12 (2015). https://doi.org/10.1145/2757001.2757003.
16. Heyvaert, P., De Meester, B., Dimou, A., Verborgh, R.: Declarative Rules for Linked Data Generation at Your Fingertips! In: The Semantic Web: ESWC 2018 Satellite Events. pp. 213–217. Springer International Publishing, Cham (2018).
17. Mader, C., Haslhofer, B., Isaac, A.: Finding Quality Issues in SKOS Vocabularies. In: Theory and Practice of Digital Libraries. pp. 222–233. Springer, Berlin, Heidelberg (2012).
18. Kitchenham, B.: Procedures for Performing Systematic Reviews.