SKOS with OWL: Don't be Full-ish!

Simon Jupp and Sean Bechhofer and Robert Stevens

School of Computer Science, University of Manchester, UK
`first.second@manchester.ac.uk`

1 Introduction

The Simple Knowledge Organisation System (SKOS)[1] is a vocabulary intended
for the representation of Knowledge Organisation Systems (KOS) such as the-
sauri, term lists and controlled vocabularies. The SKOS data model is defined
using an RDF schema so that it fits in with existing Semantic Web technolo-
gies. We speculate how SKOS artefacts may be used in combination with other
knowledge artefacts, such as ontologies represented in the Web Ontology Lan-
guage (OWL)[2] . The SKOS schema is currently OWL-Full which has implications
when using SKOS in combination with ontologies expressed in some decidable
fragments of OWL, such as OWL-DL. We demonstrate that some of the issues
can be addressed with the introduction of rich annotations and/or punning into
OWL-DL, which could be beneficial to both the SKOS and OWL communities.

Representing domain knowledge in the form of ontologies has become in-
creasingly popular in recent years. The use of these technologies in the context
of the web has been further boosted by the development of standards like the
Resource Description Framework (RDF) and OWL. In addition to representing
knowledge in the form of OWL ontologies, there also exist other knowledge arte-
facts where representation in OWL is deemed inappropriate. These KOS, which
include things like thesauri, subject heading systems and taxonomies have been
used in the library and information science domains for many years [4], [5], [3],
mostly for the purpose of query expansion and document navigation. SKOS is
being developed as part of W3C's Semantic Web Deployment Working Group
(SWDWG) activity, and Editors' Working Drafts of the SKOS reference and
SKOS Primer are now available.

This paper includes some speculative ideas. Although precise motivations for
the use cases we propose are still unclear, investigating potential patterns for
the combination of SKOS and OWL-DL does help to highlight some issues. By
doing this at an early stage, we can bring these issues to the attention of both
the SKOS and OWL communities and ensure that decisions about the future of
both standards do not rule out any potentially desirable use cases in the future.
Our discussions highlight some potential conflicts between SKOS and OWL-DL
and suggest how they could be achievable via some extensions to OWL-DL. By

---

[1] `http://www.w3.org/2004/02/skos/`

[2] `http://www.w3.org/TR/owl-features/`

a happy coincidence, the extensions proposed – namely rich annotations and punning – are currently under discussion by the OWL working group[3].

2   SKOS and OWL

The SKOS data model contains many of the standard features found in other popular knowledge organisation standards such as BS 8723 [2] and ISO 2788 [1]. The SKOS data model is described using vocabulary taken from RDF and OWL. Whilst SKOS is not a language for modeling ontologies, the data model itself is in fact described as an OWL ontology. A particular KOS is then represented as an instantiation of this ontology. SKOS Concept and Collection Scheme are defined as OWL Classes, the semantic relations and documentation properties in SKOS are Object Properties and the labeling properties are Data Properties. Some properties (for example `skos:broaderTransitive`) have additional characteristics such as transitivity and symmetry along with some relatively flexible domain and range constraints. To provide maximum flexibility, few constraints are placed on the model, leaving a number of possible use cases open for exploration. The current SKOS schema is not an OWL DL ontology, primarily due to violations of the namespace separation restrictions[4]. For many SKOS use cases this is not a problem, however, this does raises some issues when using SKOS in combination with OWL-DL ontologies.

Below we discuss three possible use cases for working with SKOS and OWL and highlight some of the current issues with respect to OWL species.

A clear use case for SKOS is as an annotation vocabulary for OWL ontologies. For example, SKOS can provide support for labeling resources using notions of preferred, alternate and hidden labels. In particular, labels on classes are useful when presenting ontologies in user facing applications. The use of standard documentation properties (such as SKOS's change notes, definition etc) would facilitate interoperation between applications. However, there are a number of issues here. The SKOS labeling and documentation properties are currently defined as OWL Object and Data properties. This is partly due to a requirement that there be subproperty relationships between some of these properties. OWL-DL only allows annotation assertions on classes which means the use of label or documentation properties on classes results in an OWL-Full ontology. Representing the properties as OWL annotation properties would preclude the possibility of extending the properties through the definition of sub properties. In some situations, the SKOS properties may be treated in both object and data contexts (for example `skos:changeNote` may be used with a simple literal value or a more complex structured object).

Figure 1 violates OWL-DL's restrictions as assertions are being made on classes. One solution would be to allow punning and to have `th:Car` as both a Class and an Individual. Alternatively, the lexical labeling properties could also be considered annotation properties, however, the extensible nature of SKOS

---

[3] `http://www.w3.org/2007/OWL/wiki/OWL_Working_Group`
[4] e.g. URIs cannot be used as simultaneously as both Object and Data Properties

```
owl:Class (th:Car)
skos:prefLabel (th:Car "Car"@en)
skos:altLabel (th:Car "Automobile"@en)
skos:definition (th:Car "A car is a....")
```

Fig. 1. Making assertions on classes with SKOS

would require support for richer annotation in OWL-DL. A third option would be to completely separate the SKOS and OWL-DL worlds. The introduction of a property `skos:as` has been proposed that could be used to add an annotation property on any OWL entity that links it to a particular SKOS concept (see Figure 2). This enables the annotation of an ontology with concepts from a SKOS knowledge base without effecting the expressivity of that OWL ontology. This proposal comes at the cost of introducing new vocabulary to SKOS and poses the question, what are the semantics of this new relationship? It also fails to solve the problem of the use of SKOS labels for OWL classes.

```
on:Car rdf:type owl:Class
th:Car rdf:type skos:Concept
on:Car skos:as th:Car
```

Fig. 2. Introduction of a new annotation property to relate an ontology class to a SKOS Concept

Expressing the SKOS model in OWL naturally provides support for extending its classes and properties.

OWL provides a platform that can be exploited to manage the construction of thesauri in SKOS. Restrictions could be applied to particular SKOS concepts in order to impose constraints on the thesaurus content and use of relationships. Figure 3 shows an example to illustrate this point, the SKOS `Concept` class has been extended with two subclasses `Model` and `Manufacturer`, the `skos:related` property has been extended with a new property called `hasManufacturer` that has its domain and range set to `Model` and `Manufacturer` respectively. We can now build a thesaurus describing car models and manufacturers and exploit OWL reasoners to enforce the correct usage of `hasManufacturer`.

```
SubClassOf (th:Model skos:Concept)
SubClassOf (th:Manufacturer skos:Concept)
SubPropertyOf (th:hasManufacturer skos:related)
Domain (th:hasManufacturer th:Model)
Range (th:hasManufacturer th:Manufacturer)
```

Fig. 3. An example of extending the SKOS data model and placing restrictions on it with OWL

Such an approach partly relies on being able to remain within a space that existing reasoner implementations can handle (e.g. OWL-DL). There are other patterns where the expressivity and semantics of OWL may be useful in building less formal structures in SKOS, for example, the use of property chains to make complex inferences between SKOS Concepts.

A final point worth mentioning is migration between SKOS like artefacts and OWL ontologies. OWL-Lite was initially seen as a simplified sublanguage of OWL, intended for building simple vocabularies and a potential format to publish KOS like artefacts on the web. However, OWL-Lite still modeled at the class level and respected the strict semantics of the sub-class relation to from hierarchical structures. Whilst tempting to convert existing thesauri into OWL T-boxes and represent the KOS structure with a subsumption hierarchy, this naive conversion is not always possible due to incompatible semantics between KOS and ontology. SKOS is now well placed to fit this gap enabling existing KOS systems to be published on the web in a format that has more appropriate semantics. In contrast, providing a thesaurus like representation of some OWL ontology may be entirely sensible for some applications, for example document navigation systems.

3   Discussion

This short paper has demonstrated some relatively simple use cases for working with combinations of SKOS and OWL. At the moment using elements of SKOS within the OWL-DL sublanguage has implications that can move an ontology into OWL-Full. Given that SKOS has a great potential as a extensible vocabulary for ontology meta-data, it is our belief that every effort should be made to ensure these two standards can be used in combination effortlessly. In the first instance, the use of SKOS labeling properties in the form of annotation properties is most desirable. The extensible nature of these labeling properties requires support for richer annotations within the OWL-DL language. Subsequent support for Class, Individual and Property punning would open up a wide ranges of possibilities for developing knowledge bases that take advantage of the functionality offered by both OWL-DL and SKOS.

References

1. Iso 2788:1986 documentation - guidelines for the establishment and development of monolingual thesauri. *Second Edition*, 1986.
2. Bs 8723-2:2005 structured vocabularies for information retrieval. *Thesauri, British Standards Institution, London*, 2005.
3. National Library of Medicine. Medical subject headings: main headings, subheadings, and cross references used in the index medicus and the national library of medicine catalog. 1st ed, 1960.
4. T. Peterson. Introduction to the Art and Architecture Thesaurus. Oxford University Press, 1994.
5. Dagobert Soergel, Boris Lauser, Anita Liang, Frehiwot Fisseha, Johannes Keizer, and Stephen Katz. Reengineering thesauri for new applications: the AGROVOC example. volume 4(4) of *Journal of Digital Information*. Oxford University Press, March 2004.