Marcia Lei Zeng (Kent State University)
Michael Panzer (OCLC, Dublin, Ohio)
Athena Salaba (Kent State University)

Expressing classification schemes with OWL 2
Exploring issues and opportunities based on experiments using OWL 2 for three classification schemes

Abstract
Based on the research on three general classification schemes, this paper discusses issues encountered when expressing classification schemes in SKOS and explores opportunities of resolving major issues using OWL 2, Web Ontology Language.

1: Background
The year 2009 has witnessed the new releases of two major encoding languages for knowledge organization systems (KOS): SKOS: Simple Knowledge Organization System reference (Miles & Bechhofer 2009), a W3C recommendation released in August, and OWL 2: Web Ontology Language structural specification and functional-style syntax (Motik et al. 2009), a W3C proposed recommendation as of 22 September, along with a set of OWL 2 documents. The releases reflect major changes implemented in these two important encoding languages as a result of feedback received from their world-wide use cases studies. The releases also present a great opportunity of expressing classification systems with these universal encoding languages.

Classification systems have existed for a long time and have been modernized for use in all possible media and/or formats; from print to machine-readable, Web-browsable, and now machine-processable. With the linked data movement showing significant progress and solid achievements, the library and information community also made thesauri and subject headings available as linked data. Enabling classification systems to become part of linked data and using them to better organize linked data has attracted new attention in the knowledge organization communities and beyond as well. The question still remains whether classification systems can follow the example of thesauri and subject headings to encode the schemes with SKOS and if there are different issues for classification systems.

SKOS defines classes and properties with the aim to be sufficient for representing the common features found in a KOS such as thesaurus, taxonomy, controlled term lists, classification, and others. As an application of the RDF (Resource Description Framework), SKOS allows concepts to be composed and published on the World Wide Web, linked with data on the Web, and integrated into other concept schemes. “Using SKOS, concepts can be identified using URIs, labeled with lexical strings in one or more natural languages, assigned notations (lexical codes), documented with various types of notes, linked to other concepts and organized into informal hierarchies and association networks, aggregated into concept schemes, grouped into labeled and/or ordered collections, and mapped to concepts in other schemes” (Miles & Bechhofer 2009, Synopsis, emphasis added). Although SKOS provides the means to encode notations for classification systems, it is fundamentally designed to support the structure of non-classification systems such as thesauri.
Since the last ISKO Conference in Montreal in 2008, the authors have been exploring the issues involved in the expression of enumerative and partly-enumerative classification schemes with the 2008 version of SKOS (Panzer 2008; Zeng & Fan 2008). Detailed comments were submitted to the W3C Semantic Web Deployment Working Group and SKOS editors in 2008. These issues were also reported in a Decimal Classification Editorial Policy Committee (EPC) exhibit (EPC 130-37.2.2) in 2009. The major issues included the following:

- Non-assignable concepts: e.g., “centered entries” in the DDC which are number spans that form an exception to the regular notational hierarchy.
- Multiple index terms associated with one given class.
- Class and the topics that are associated with that class by class-topic relationships.
- Semantically meaningful order in a classification system.
- Auxiliary tables.
- Notation building conventions.
- Classification notations and captions.
- Mappings between classification schemes and between classification and other structures (such as thesauri).

Several questions related to the specific issues raised by the authors resulted in no change to the 2009 SKOS specification. All above issues submitted have been closed, mainly because, among other reasons, classification systems were not of major concern in the original SKOS use cases and requirements (W3C 2007). The authors then experimented with local extensions of SKOS, including extensions for class expression, notations, and mapping (Panzer & Zeng 2009). Nevertheless, even with the extensions many issues remain unsolvable within the current SKOS semantic framework.

The authors, realizing the differences between the underlying models SKOS and OWL (Web Ontology Language) are intended to support, turned to OWL 2. OWL 2 has a W3C proposed recommendation status and has ensured backward compatibility with OWL specification released in 2004 (called OWL 1) to explore issues and opportunities. OWL ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents (OWL 2 Document Overview, 2009). OWL 1 focused mainly on constructs for expressing information about classes and individuals. OWL 2 offers new constructs for expressing additional restrictions on properties, new characteristics of properties, incompatibility of properties, properties chains, and key properties (W3C OWL Working Group 2009).

2: Exploring OWL 2 in expressing classification schemes

Targeting on specific issues of classification, three general classification schemes are used in our experiments: the Dewey Decimal Classification (DDC), the Chinese Library Classification (CLC) (4th ed. 1999; 5th ed. in progress), and the Library of Congress Classification (LCC). DDC and CLC have a very similar structure which was designed primarily as enumerative, but has moved increasingly towards synthetic features to allow forming access numbers by following rules for “building” and “adding”. DDC and CLC both have numerous tables embedded in subject schedules as well as several auxiliary tables. The index is another component. Both classification systems have theoretical base for the sequence of disciplines. LCC, in comparison with other general classifications, represents the most enumerative scheme. It provides for some synthesis,
and some schedules have tables such as geographic tables and form divisions. LCC uses a pragmatic approach instead of grounding on a theoretical basis. In addition, other special classification schemes and ontologies designed for specialized domains are also consulted in this study.

OWL ontologies and traditional classification systems share a great deal of common characteristics, especially in terms of presenting various kinds of classes and relationships of classes. Classes in OWL can be understood as sets of individuals. OWL 2 has added a new syntactic subset in a profile to accommodate ontologies needing to represent rather complex entities. OWL ontologies exhibit a huge number of classes and have a heavy use of classification to manage their terminology (Golbreich & Wallace 2009). These features make OWL 2 appropriate for encoding classifications and solve many issues discussed in the following section.

In OWL 2, classes and property expressions are used to construct class expressions and complex concepts. The most useful characteristics used in dealing with the issues of classification are summarized below (all references are from Motik et al. 2009):

1. OWL 2 supports various ways of describing classes: class identification, the intersection and union of two or more class descriptions, the complement of a class description, property restrictions, and the enumeration of individuals that form class instances. Complex ClassExpressions include:
   - ObjectIntersectionOf | ObjectUnionOf | ObjectComplementOf | ObjectOneOf
   - ObjectSomeValuesFrom | ObjectAllValuesFrom | ObjectHasValue | ObjectHasSelf
   - ObjectMinCardinality | ObjectMaxCardinality | ObjectExactCardinality

   All standard Boolean connectives AND, OR, and NOT are supported. The ObjectIntersectionOf, ObjectUnionOf, and ObjectComplementOf provide for the standard set-theoretic operations on class expressions. The ObjectOneOf class expression contains exactly the specified individuals.

   Class expressions in OWL 2 can be formed by placing restrictions on object property expressions. For example, the ObjectSomeValuesFrom allows for existential quantification over an object property expression, and it contains those individuals that are connected through an object property expression to at least one instance of a given class instance.

   Class expressions in OWL 2 can be formed by placing restrictions on the cardinality of object property expressions. Cardinality restrictions can be qualified or unqualified. The class expressions ObjectMinCardinality, ObjectMaxCardinality, and ObjectExactCardinality contain those individuals that are connected by an object property expression to at least, at most, and exactly a given number of instances of a specified class expression.

   All of these are especially useful when representing various types of classes, regardless if these classes are already established in a classification scheme or instructed to synthesize in the classifying process. Various class expressions can be used to precisely express the situations where, for example, values from one or more auxiliary tables are allowed to be used; or, in certain cases, if some values from a class can be added to another class.

2. OWL 2 provides axioms (statements that say what is true in the domain) that allow relationships to be established between class expressions, including the following:
   - SubClassOf axiom: allows one to state that each instance of one class expression is also an instance of another class expression, and thus to construct a hierarchy of classes.
   - EquivalentClasses axiom: allows one to state that several class expressions are equivalent to each other.
   - DisjointClasses axiom: allows one to state that several class expressions are pairwise disjoint — that is, they have no instances in common.
   - DisjointUnion class expression: allows one to define a class as a disjoint union of several class expressions and thus to express covering constraints. Such axioms are sometimes referred to as covering axioms.
Subclass axioms are a fundamental type of axioms in OWL 2 and can be used to construct a class hierarchy. This would be most widely used in a classification scheme. Other axioms can solve special problems, e.g., the “alternative class position” issue (refer to next section).

(3) OWL 2 supports two kinds of object property expressions. Object properties are the simplest form of object property expressions, and inverse object properties allow for bidirectional navigation in class expressions and axioms. The inverse object properties would be especially useful for the expressions of a class and its related index entries (refer to next section). Object property expressions can be employed to represent various kinds of rules for building classification numbers.

The authors have investigated classification issues in three main categories: (1) a classification scheme’s structure and components, (2) classes and their relationships, and (3) notations. Selected elements from the three classification schemes are transformed to the workspace using the Protégé OWL editor. The following text provides a general discussion of the issues encountered when experimenting in SKOS and further tested in OWL 2. Due to limited space in this publication, screenshots and detailed explanations are given at the presentation.

3: Issues in presenting various types of classes and relationships of classes

Explicitly listed classes. Classification schemes are designed to describe and structure all types of subjects, whether simple, compound or complex. There are no problems in expressing subjects explicitly listed in a classification’s schedule when using SKOS.

“Centered entries” (or other kinds of number spans). One special situation in a classification system is the “number spans” or – in case of the DDC – so-called “centered entries” as a specific type of span. Although non-assignable when assigning classification numbers, they are not included in a system just as a display or a presentation device; rather, they are an integral part of the system. Centered entries relate notationally coordinate (i.e., sibling) classes together as a single class in cases where a notation is not available for use in the hierarchy. In the following example, T2-485 represents Sweden; the centered entry T2-486–T2-488 represents the geographic divisions of Sweden. These would be hierarchically subordinate to T2-485. Nevertheless, they occupy numbers that are coordinate to the T2-485 and represent a true broader concept. A new class in SKOS or expanded skos:Collection classes would be required to allow concept collections like “number spans” or “centered entries” to be expressed as concepts.

Synthesis in classification schemes. To accommodate new subjects and reflect the nature of interdisciplinary domains, primarily enumerative classification schemes have used various approaches for synthesizing numbers or building new class numbers. For example, through post-coordination, new numbers are formed in practice (not in the classification scheme) by combining elements from different parts of the structure to construct a number representing the subject content. Issues involved range from expressing compound and complex subjects to presenting the relationships between these classes and others. Again, these bring additional requirements to SKOS.

Class-topic relationships. This issue is related to the distinction between a class and the topics that are associated with that class. Topics are gathered and extended by various mechanisms into what could be called the topical neighborhood of a class, thus
forming part of its category description. The semantic space of a topical neighborhood
takes the form of a graph of topics that is connected with a class by a set of class-topic
relationships (Green & Panzer 2010). In order to adequately model a classification
system, we have to recognize not only the class, but also the topics (broadly construed)
associated with the class, enabling us to assert relationships not only between classes,
but also between classes and topics. With SKOS as a representational model, it becomes
difficult to make that distinction (Panzer & Zeng 2009) without restricting the
possibility to assert intra-scheme relationships to only the class level. Treating class-
topic and class-class relationships the same way (i.e., as relationships between classes)
might produce inconsistencies with the SKOS model (i.e., classes, regarded as
skos:Concepts, may end up being connected by hierarchical and non-hierarchical
relationships at the same time). OWL, however, opens up the possibility to make that
distinction early on by creating two primitive ontological classes, with topics of the
scheme being instances of one, classes of the scheme being instances of the other. These
two ontological classes are disjoint, as no individual of the domain can be in the
extension of both a topic and a class of the scheme.

Alternative class location. Enumerative (rather than faceted) structures often face the
issue of interdisciplinary topics/classes. “Environmental biology”, for example, can be
listed in both “biology” and “environmental science” classes. An alternative notation,
designed for these situations, provides the flexibility to place the class in all applicable
contexts to ensure systematic presentation of topics/classes in a discipline or domain.
Regardless of whether it is the preferred or alternative, a notation always represents a
unique concept with different semantic relationships. Therefore, an alternative notation
does not resemble a non-preferred thesaurus label that has only lexical relationships.
This requires a very different treatment compared to a thesaurus, where different labels
of one concept can be treated as semantically equivalent.

Presenting auxiliary tables. An auxiliary table, as a whole, has a top concept. It
should be specifically expressed as both a top concept of a table and a member of the
whole classification scheme. This goal can only be partially achieved using SKOS with
the addition of local extensions. Another issue is that each subdivision in a table can be
both a member of the auxiliary table and a part of a synthesized class number when
combined with any class number in a subject schedule.

Presenting index entries. An important part of a classification system is the index
(a.k.a. “Relative index” in DDC). Index terms associated with a given class generally
reflect several topics falling within the scope of that class. These indexes can be very
substantial in size (e.g., the CLC’s 4th edition index has over 120,000 entries) and may
be more complex than some independent thesauri. Currently, a possible workaround for
encoding in SKOS is to construct a complete index as a separate
skos:ConceptScheme. Then, the concepts in the index can be related to those in the
subject schedules, which is also a skos:ConceptScheme, by using mapping relations.
In OWL 2, inverse object properties allow for bidirectional navigation in class
expressions and axioms and should be able to deal with the index issue.

Presenting order/sequence of sibling classes. Hierarchical relationships between
concepts have been well handled by all major representational languages including
SKOS. Through skos:broader one can trace to the top concept based on the explicit
expression of the relationships between and among concepts. However the relationships
of coordinates (sibling classes or concepts) are not expressed in SKOS. Classification systems have a tradition of producing orders that are semantically meaningful. Such an order is evident in the juxtaposition of classes, the sequence of main classes, and the sequence of coordinates within a class. The choice of sequence is usually made on the basis of an underlying principle. From another point of view, a notation should convey both a semantic and an ordinal value. The semantic value of a classification number is the subject or concept it stands for. In SKOS this is carried out through `skos:notation` for a given concept. The ordinal value of a number or code places the subject into its determined rank in the scheme; and such a value has not yet been considered in SKOS. Although SKOS provides for ordered collections to be used in conjunction with `skos:memberList` to express specific sequencing of concepts, the `skos:Collection` is disjoint with `skos:ConceptScheme`.

**Internal structure of notes.** Classifications usually provide notes that contain instructions or references associated with a class or its hierarchical array. The functions of these notes are so diverse that documentation properties in SKOS can only serve as an extension point for further specification. OWL 2 has a variety of ways to deal with these notes and reveal their content effectively.

**Presenting notation-building rules.** Each classification implements certain rules for building notations. The following are some typical examples. (1) In current authority systems there are always records indicating how a notation should be composed or decomposed. (2) A synthesized number can be constructed by adding or appending numbers from a table or from other parts of the schedule. Instructions are provided to the classifier to construct a classification number by adding numbers from other parts of the schedule, from a table, or by basing it on a pattern defined in another part of the schedule. (3) Depending on the degree of synthesized components, some classification schemes have a variety of faceted structures for their main schedules, sub-schedules, or individual classes. Rules and instruction guidance are always included in such cases. (4) There could be full, abridged, and extended (+) notations for the derivations from a general classification system. Unlike a thesaurus, a classification system usually develops variations of a scheme with different scales. Implementers decide to which degree they want to implement it in practice. For example, in CLC’s 062.32+6 the number after + is the optional, extended number with higher specificity.

There are additional situations and issues that the authors are currently exploring. Evidence of using OWL 2 to resolve some major issues related to classification systems is emerging. This evidence may also lead us to consider the differences of SKOS and OWL that mainly support two different kinds of models represented by thesauri and classification systems.
References