Modeling Classification Systems in SKOS: Some Challenges and Best-Practice Recommendations

Michael Panzer
OCLC, USA
panzerm@oclc.org

Marcia Lei Zeng
Kent State University, USA
mzeng@kent.edu

Abstract

Representing classification systems on the web for publication and exchange continues to be a challenge within the SKOS framework. This paper focuses on the differences between classification schemes and other families of KOS (knowledge organization systems) that make it difficult to express classifications without sacrificing a large amount of their semantic richness. Issues resulting from the specific set of relationships between classes and topics that defines the basic nature of any classification system are discussed. Where possible, different solutions within the frameworks of SKOS and OWL are proposed and examined.

Keywords: classification systems; knowledge representation; Simple Knowledge Organization System; Web Ontology Language; Dewey Decimal Classification; Chinese Library Classification

1. Introduction

The Simple Knowledge Organization System (SKOS), as it is approaching W3C recommendation status, promises to be the most important RDF vocabulary for the publication of knowledge organization systems such as thesauri, taxonomies, and classification schemes as elements of the Semantic Web. Yet the use cases and requirements for SKOS have been mostly focused on thesaurus-like vocabularies or classifications that have a relatively simple internal structure. To effectively publish classification systems with SKOS, interoperable extensions and common best practices will have to be developed by the community in order to retain the usefulness of representing a classification in SKOS. The Semantic Web stack (Berners-Lee, 2000) provides a robust foundation and a set of tools to extend what has already been developed by SKOS as a general core, and also to reduce the added complexity depending on the processing abilities of the user agent. Since all extensions are derived from SKOS or other RDF vocabularies in a stringent manner, a user agent is able to tailor its view of the data in accordance to its reasoning capabilities or knowledge of particular metadata elements.

In the context of experimenting with possibilities of implementing SKOS, the authors have been exploring issues involved in the expression of large classification systems. Since Linked Data initiatives put a strong emphasis on representing KOS for use cases involving discovery and access (Berners-Lee, 2007), we tried to keep the suggested models as simple as possible, but without sacrificing the richness classification systems provide to users for resource description and discovery.

2. Specific representational issues

2.1. Special types of concepts

Classification systems usually contain objects that, while not being assignable concepts, are nonetheless an integral part of the system (not just a display/presentation device), e.g., number spans or—in case of the DDC—so-called "centered entries:"

T2—486–T2—488 Divisions of Sweden

333.7–333.9 Natural resources and energy

https://doi.org/10.23106/dcmi.952109518

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. For example, T2—485 represents Sweden; the centered entry T2—486–T2—488 represents the geographic divisions of Sweden which are hierarchically subordinate to T2—485, but occupy numbers that are coordinate to the number that represents the broader concept Sweden.

Centered entries are an important part of the structural hierarchy of the DDC. They represent true broader concepts, even though this superordination is indicated by other devices than directly by a shorter number. On the one hand, because they often also contain instructions applicable to all subordinate classes, centered entries cannot be modeled as a skos:Collection. As defined in the SKOS Reference (2009, sec. 9.6.4), skos:Collection and skos:Concept are disjoint. Meanwhile, the domain and range of semantic relationships in SKOS are restricted to skos:Concept. As a result, a skos:Collection cannot be part of a concept hierarchy that is established by the use of skos:broader or skos:narrower, as they both count as semantic relationships.

Another attempt would be to expand the skos:Collection class to accommodate these kinds of concepts. This does not seem to be appropriate either, because the grouping function that centered entries perform is very distinct from that of true concept collections that exist in classification systems, e.g., auxiliary tables (to be discussed later on).

If spans or centered entries are not collections but a way in which members of a broader concept can be partitioned into narrower concepts, then a new SKOS class is required that allows them to be expressed. A subclass of skos:Concept should be defined, e.g., skosclass: NonAssignableConcept, with an additional cardinality constraint (using owl:cardinality from the OWL Web Ontology Language, 2004) that prohibits the use of an indexing property for that class. Since all indexing properties have been dropped from SKOS, a property would have to be picked arbitrarily (like dct:subject from the DCMI Metadata Terms) that is likely to being used in applying the classification.

The definition of new concept types nevertheless adds considerable flexibility to modeling classifications on the basis of SKOS. It effectively allows further semantic specification of the new types using OWL class descriptions beyond cardinality constraints. As classification systems contain a variety of concepts that require or would benefit from special definitions, we propose a set of skos:Concept subclasses as an extension of SKOS to better accommodate the needs of classification systems (see Figure 1).

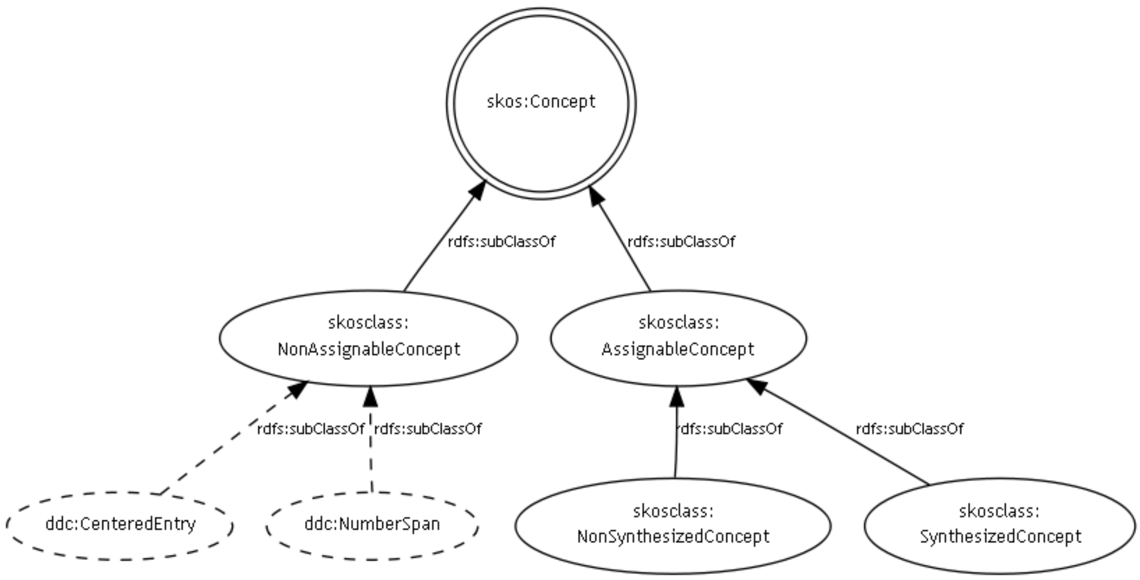

FIG. 1. Extended concept types.

https://doi.org/10.23106/dcmi.952109518

To further distinguish non-assignable from assignable concepts, a second subclass skos-class:AssignableConcept is created that is disjoint from its sibling class. Many classification systems contain assignable and non-assignable classes, but the nature of non-assignable classes is more diverse than that of assignable classes. It therefore seems beneficial to further specify the class of assignable concepts to accommodate non-synthesized and synthesized concepts (i.e., classes that have been combined by using other concepts of the classification), but leave the refinement of non-assignable concepts to local extensions (e.g., centered entries vs. number spans in case of the DDC).

Major drawbacks of this approach in practice are that some user agents would not be prepared to deal with these extensions and that classes that are specifications of skos:Concept could not be retrieved with one simple query or search strategy. There is currently no support in SPARQL (the query language for RDF graphs, see *SPARQL Query Language for RDF*, 2008) for querying transitive relations and generally very limited support in RDF frameworks for inference, so the burden of "dumbing down" or retracing the extensions to the elements they were derived from generally falls to the user agent.

Collections are not entirely out of the picture, however. We already assumed above that collections might be more appropriate to accommodate another special device that many classification systems share: auxiliary tables. Although SKOS does not handle such components within a scheme directly, auxiliary tables can be expressed as a skos:Collection without major difficulties. The following example defines Table 2 in the DDC as a concept collection and allows for further documentation using lexical, note, and even notation properties provided by SKOS. This and other RDF examples are presented using the Turtle syntax (developed David Beckett & Tim Berners-Lee, 2008).

```
<table/2/> a skos:Collection ;
    skos:prefLabel "Geographic Areas, Historical Periods, Persons"@en ;
    skos:prefLabel "Geographische Gebiete, Zeitabschnitte, Personen"@de ;
    skos:member <class/2--485/> .
```

It is not possible, however, to define separate top concepts for collections, as the domain of skos:hasTopConcept is restricted to skos:ConceptScheme (that is disjoint with skos:Collection). To retrieve the top concept(s) of an auxiliary table, a user agent would have to search for classes that are both top concepts of the scheme and members of that auxiliary table. By using nested collections (e.g., defining a collection that contains all separate collections of table classes, therefore being able to retrieve all top concepts of all tables) it is possible to establish top concepts in an even more hierarchical way (Figure 2).

https://doi.org/10.23106/dcmi.952109518

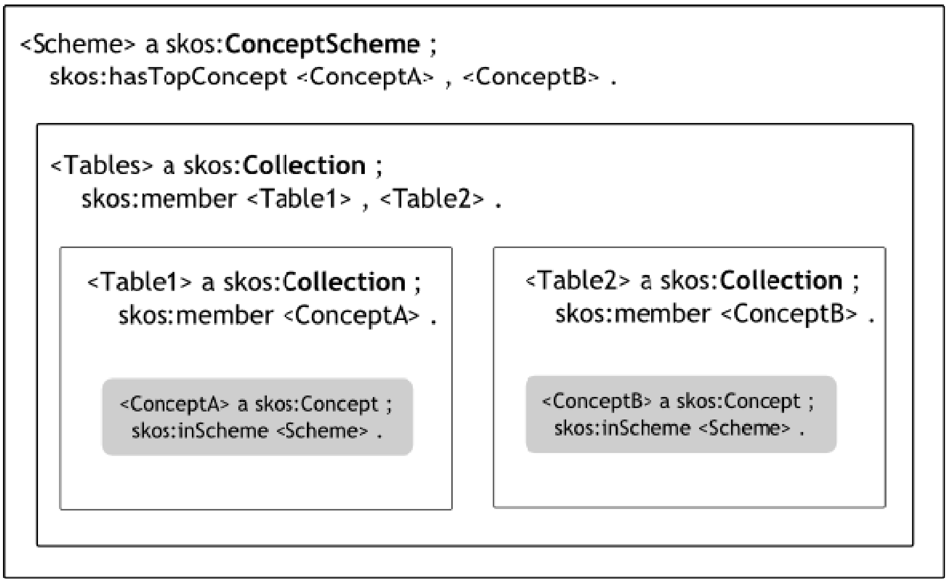

FIG. 2. Nesting of collections in combination with assertions of top concepts allows for random access to top concepts of collections.

Yet this practice seems to go slightly against the grain of the SKOS data model, because skos:ConceptScheme is on the one hand defined as an aggregation of skos:Concepts, but on the other hand is disjoint with skos:Collection; a collection can never be asserted to be part of a concept scheme. Therefore, it does not seem ideal to transfer too much structural information about scheme–concept relations into the nesting of collections (in order not to blur the scope of these two disjoint elements).

2.2. Index terms

An important part of many classification systems is their index, e.g., the "Relative Index" of the DDC or the "General Alphabetical Index" of the iconographic classification system ICONCLASS (the latter index spans three volumes in the printed version). The index is a major feature especially of larger classification systems because—even in its simplest form as an alphabetical keyword list—it is often used as access vocabulary for guiding the classifier to the appropriate place in the classification scheme or for providing terminology for subject description and retrieval. These indexes can be very substantial in size and may be more complex than many independent thesauri.

Index terms associated with a given class generally reflect several of the topics falling within the scope of that class, yet there is no easy way of modeling this relationship in SKOS. For example, the Dewey class 616 Diseases has the following index terms:

Clinical medicine
Diseases—humans—-medicine
Illness—medicine
Internal medicine
Physical illness—medicine
Sickness—medicine

Currently, a possible workaround is to construct the complete Relative Index as a separate skos:ConceptScheme and to relate the concepts in these two independent schemes by using mapping relations:

https://doi.org/10.23106/dcmi.952109518

```
skosclass:hasIndexTerm rdfs:subPropertyOf skos:closeMatch .
skosclass:isIndexTermOf rdfs:subPropertyOf skos:closeMatch ;
    owl:inverseOf skosclass:hasIndexTerm .
<class/616/> a skos:Concept ;
    skosclass:hasIndexTerm <index/Clinical%20medicine/> ;
    skos:inScheme <classification/> .
<index/Clinical%20medicine/> a skos:Concept ;
    skosclass:isIndexTermOf <class/616/> ; skos:inScheme <index/> .
```

This effectively shifts the focus to that of linking/matching the two independent instances of skos:ConceptScheme. In addition, this strategy opens up the possibility of defining different types of top concepts (one for the index, one for the classification) without using skos:Collection or deriving subproperties of skos:hasTopConcept for just that purpose. It seems to be a satisfactory best-practice solution in this case, but it has broader implications, as index terms are generally just one instance of class–topic relations (to be discussed further in the next section) that are of major importance for all classification systems.

2.3. Class–topic relationships

Classes usually don't form the atoms of a classification system. Either implicitly or explicitly, a classification recognizes topics that fall into the neighborhood established by a class as part of its fundamental structure. Since SKOS only allows modeling of lexical relationships at the concept level, reflecting this set of relationships is essentially out of the realm of SKOS. Even extensions using OWL constructs cannot effectively help here. Whenever classes are treated as instances of skos:Concepts they become individuals of the domain, restricting the ability of expressing relationships to the class level (in the sense of classes in a classification systems). OWL language constructs that operate on classes (in the ontological sense) cannot be applied to them as individuals unless leaving OWL-DL for a more expressive language like OWL Full, precluding the use of reasoners or inference engines.

This issue seems to cause some general problems for using SKOS as a universal tool to model classification systems, since the relationship between topics and classes acts as an important force for shaping concepts in a classification system. There are numerous examples of problems that arise from the difficulty of expressing in SKOS the interplay between a class and the topics that, on the basis of similar characteristics, form the semantic space of that class.

The inability to model other than concept–concept relationships with SKOS sometimes leads to inconsistencies as subjects/topics are frequently in the domain or range of common classification relationships.

In the DDC, this can manifest itself in classes being connected by both hierarchical and non-hierarchical relationships if modeled with current SKOS semantic relations:

```
<A> skos:narrower < B> . <B> skos:related <A> .
```

This inconsistency with the SKOS data model arises because what is expressed here isn't really a relationship between classes, but between topics and classes.

```
<A> skos:narrower <B> . <TopicInB> skos:related <A> .
```


https://doi.org/10.23106/dcmi.952109518

This pattern can also lead to circular hierarchical relationships:

<A> skos:narrower <TopicInB> . <B> skos:narrower <TopicInA> .

At the moment, there is no other way in SKOS than to code these relations at class level:

<A> skos:narrower <B> . <B> skos:narrower <A> .

This results in obvious inconsistencies. A possible solution would be to introduce/define ddc:related (or similar relationships) as a new element without extending semantic relationships provided by SKOS, even if this means lowering the utility of classification systems in SKOS applications, as no simpler SKOS relationship could be inferred by a user agent.

Refining the model described above, it has been suggested that topics be expressed as instances of skos:Concept belonging to a separate concept scheme. SKOS allows the inter-scheme use of semantic relations like skos:broader, so these new concepts can be integrated into the general classification hierarchy. However, this approach does not take into account that the topics a class corresponds to are not necessarily exhaustively enumerated by the classification system. The set of topics that forms the neighborhood – the semantic space – of a class is defined, extended or restricted by the complex interplay of different notes, application rules, and even classification practice.

Beyond these techniques, there does not seem to be an adequate solution to this problem using the SKOS model. We speculate that the expressiveness of OWL, when used independently of SKOS, provides the desired degree of expressiveness, as its semantic formalization of class/subclass relations combined with the flexibility of class descriptions to constrain/define class extensions seems to correspond to a larger degree with the class–topic model of classification systems.

2.4. 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 (note properties) in SKOS can only serve as extension points that need further specification when applied to classification systems. Although the handling of notes in general is not discussed in this paper, nowhere else is the general topical nature of classification systems more apparent than in modeling a specific note type: the history of a class.

The association of a topic with a class might change frequently as classes are expanded, discontinued, or topics are relocated. SKOS provides a few subproperties of skos:note to attach history information to, and Tennis and Sutton (2008) have suggested that SKOS be extended on the concept level with an additional concept type to improve the management of historical continuity of SKOS schemes. In order to make the history of a class useful for retrieval interactions (that frequently depend on determining which class the topic was associated with at a given time) further specification of the structure of skos:historyNote is necessary. A full discussion of the complex implications of managing and exposing ontogenesis of classes is outside the scope of this paper, but the general structure of a history note in the DDC can usefully illustrate the specific challenges that many classification systems share in translating the meaningful structure of notes into machine-processable semantic constructs.

The extensibility of SKOS and the flexibility of RDF allow for the mixing of properties from different vocabularies, including newly created ones. The following example combines properties from SKOS itself with properties and datatypes from the DCMI Metadata Terms, plain RDF and new properties declared in the "ddc:" namespace that are in part semantic extensions of SKOS.

https://doi.org/10.23106/dcmi.952109518

As extension point skos:historyNote was chosen (over skos:changeNote) because the changes indicated here (or in similar notes) are highly relevant to users and user agents alike, potentially driving retrieval interactions and collection management. This matches closely the intended use described in the SKOS Primer (2009, sec. 2.4): "skos:historyNote describes significant changes to the meaning or the form of a concept." By using a resource description instead of an RDF literal, additional and more structured information can be conveyed. At the same time, the textual content outlining the change is still documented as an rdf:value.

```
ddc:formerlyNote rdfs:subPropertyOf skos:historyNote .
<class/572/> rdf:type skos:Concept ;
    skos:notation "572"^^<schema-terms/Notation> ;
    skos:prefLabel "Biochemistry"@en ;
    ddc:formerlyNote [
        dct:issued "1996-01-01"^^<http://purl.org/dc/terms/W3CDTF> ;
        dct:isPartOf <scheme/e21/> ;
        dct:description "Class immediately reused"@en ;
        ddc:previousNumber "574.192"^^<schema-terms/Notation> ;
        rdf:value "Biochemistry formerly located in 574.192"@en ;
        ddc:topic "Biochemistry"@en .
    ] .
```

Making history information available in this slightly formalized manner allows for exploratory SPARQL queries to answer questions that could not have been answered before. The information that, before 1996, this class used to have a completely different meaning (which is shown by another note), that biochemistry used to be located in 574.192, that this change happened in Edition 21 of the DDC, etc. used to be hidden in the history note; now it is accessible to machine-processing. Ideally, the topic relation expressed in the example above would use a URI reference, not a literal, but that requires a more stable approach to the modeling of class–topic relationships as outlined above.

It should be noted, however, that using skos:historyNote in this way instead of attaching the history information directly to the concept URI as the subject (1) introduces a blank node into the graph and (2) changes the semantics of some triples established by the history note slightly. Arguably, the subject of ddc:previousNumber should not be the history note itself as it is in the model shown above, but the topic or sometimes even the concept in question. Using the concept as subject for history-related statements would lose the convenience of having a single piece of history information gathered together as one resource. The other alternative, i.e., using the topic as subject, is not possible until the topic gets its own identifier and becomes a resource; a literal can never be a subject in an RDF statement.

2.5. Alternative classification notations

In classification systems which are based on hierarchical (rather than faceted) and enumerative structures, there is always the contextual issue of which discipline a class should be located in and how interdisciplinary treatment of a topic should be handled. Environmental biology, for example, can be listed in both biology and environmental science classes. A traditional classification system, however, often has to allow the choice of only one location when the scheme is adopted by a particular library for physical location.

In the case of the *Chinese Library Classification* (CLC), on some occasions an alternative (i.e., optional) notation is given for a concept. Such kind of alternative notation is indicated by a

https://doi.org/10.23106/dcmi.952109518

bracket. For example, an Alternative notation (Q89) "Environmental biology" mentioned in the above example is one of such cases appearing in the first level subdivisions of the CLC. For example:

[Q89] 环境生物学
宜入X17。

Q Biological sciences
… …
[Q89] Environmental biology
Preferred class: X17

X17 环境生物学
环境生物工程入此。
X171 生态系统与污染生态学
环境生态学入此。
参见S181。
X171.1 生态系统与生态环境
[X171.3] 生态农业
宜入S181。
X171.4 生态建设与生态恢复
生态工程的研究入此。
X171.5 污染生态学
生态毒理学入此。
X172 环境微生物学
微生物降解和转化污染物的规律入此。
X173 环境植物学
植物毒性、树木园林与环境保护的关系入此。
X174 环境动物学
动物毒性研究入此。
X176 生物多样性保护
参见Q16。

X1 Environmental Sciences – Basic Theory
… …
X17 Environmental biology

Note: Here under **X17 Environmental biology**, all subdivisions and semantic relationships between a class and other classes are systematically presented.

If in an implementation a decision is made to use Q89 as the preferred class and X17 as the alternative class, the bracket will be moved from Q89 to X17. The subdivisions under Q89 will be formed following those listed under previous X17. The semantic relationships of those classes will be kept.

FIG. 3. Alternative notation [Q89] and its equivalent, preferred class X17.

There are many other cases where such practices are implemented, such as that occurred in Main Class X Environmental Science (Figure 4).

[X18] Environmental medicine →
Preferred class: R12

[X191] Environmental psychology →
Preferred class: B845.6

[X197] Environmental law →
Preferred class: D912.6

[X18] 环境医学
总论宜入R12。
专论宜入R其他有关各类。例：总论环境医学入R12;职业性疾病预防入R135；环境流行病学入R18；地方病学入R599；环境毒理学入R994.6。
[X191] 环境心理学
宜入B845.6。
X192 环境系统学（环境系统工程）
X196 环境经济学
参见F062.2。
[X197] 环境法学
宜入D912.6。

FIG. 4 Examples of alternative classes in the Chinese Library Classification (CLC).

https://doi.org/10.23106/dcmi.952109518

Regardless of whether it is preferred or alternative, the notation always represents a unique concept and therefore has semantic relationships with other concepts. If placed in environmental science, the concept "environmental biology" that has the notation "X17" is the subdivision of "X1 Environmental Science – Basic Theory." The other (alternative) notation is Q89, which has a broader concept "Q Biological Sciences."

With the current SKOS vocabulary, there is the option of using multiple notations for such a concept since SKOS does not place cardinality restrictions on skos:notation. To distinguish between preferred and alternative notation, it would be necessary to specify skos:notation as skosclass:altNotation. But emphasizing the alternative nature of one notation over the preferred notation does not address the real issue here, which results from the fact that SKOS separates concepts from their lexicalizations. Here, however, regardless of whether it is preferred or alternative, the notation always represents a unique concept with different semantic relationships. This requires a very different treatment compared to a thesaurus where different labels or notations of one concept can be treated as semantically equivalent.

In a classification system like the CLC, however, depending on the choice of notation, a concept is placed with a different set of classes that share common attributes and has its own semantic relations. Hence, an alternative notation is not a non-preferred thesaurus label that has only lexical relationships.

Although there is no easy remedy available in SKOS for this conceptual difference, one way to conceptualize this intertwining of lexical and semantic properties is (following the SKOS extension for labels) to transform the alternative notations into resources (instead of literals). As resources, they can possess different semantic relationships:

```
<X17> skos:broader <Environmental Science_Basic Theory> .
<Q89> skos:broader <BiologicalSciences> .
<Environmental Biology> clc:altNotation <X17> .

skos:broader owl:propertyChainAxiom  ( clc:altNotation skos:broader ) .
```

The connection back to the original resource is established by the definition of a property chain axiom. On the basis of this axiom, an OWL reasoner can infer that if the concept "Environmental Biology" has the notation "X17," and "X17" has as a broader concept "Environmental Science – Basic Theory," then "Environmental Biology" has as a broader concept "Environmental Science – Basic Theory." Based on the assertion of a notation, a semantic relationship is propagated along our new notation property.[1]

2.6. Order in Classification Systems

According to Svenonius, "the creation of a meaningful order is equally as important in information organization as the grouping of documents into classes" (2000, p. 191). Classification systems have a tradition of producing orders that are semantically meaningful. Nevertheless, unlike hierarchical relationships that have been explicitly expressed through all major representational languages for KOS such as SKOS and OWL, the coordinates – sibling classes/concepts – have attracted less attention among the editors of those standards.

[1] Because of domain/class restrictions in SKOS, this approach works only if we define "X17" to be an instance of skos:Concept. Otherwise, we would have to create our own set of semantic relationships to accommodate the specific nature of this extended notation resource.

https://doi.org/10.23106/dcmi.952109518

A semantically meaningful order in a classification system is critically important. It is evident in the juxtaposition of classes, the sequence of main classes, and the sequence of co-ordinates in a class. The choice of sequence is usually made on the basis of an underlying principle. Common arrangements of coordinate classes may be based on one or more of the following principles:

- arrangement by stages in a process (e.g., brewing processes, packaging of products);
- arrangement by time or evolutionary sequence (e.g., ancient Greek sculptures, paleontology, stars);
- arrangement by degree of complexity (e.g., geometric figures);
- arrangement by size (e.g., town, cities, metropolis, and other administrative units);
- according to principles of literary warrant (e.g., arrangement of literature according to publication amount);
- according to principles of user warrant (e.g., arrangement of services and products according to popularity).

Notation is the system of codes attached to concepts and subjects in a classification scheme. Any notational scheme has both a semantic and an ordinal value (Iyer, 1995). The semantic value of a classification number is the subject or concept it stands for. In SKOS this is handled 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. For example, it is natural to arrange library materials of "Biology" before "Medicine," given the nature of the knowledge structure in related disciplines. As a result, the notations assigned to these classes must reflect such an order. However, in SKOS such an ordinal value has not yet been considered.

One argument is that the notation's value itself implies the position of a class or concept. This is partially true if a computer system will be able to reason based on the comparison of *values of notations* or be able to reverse the whole ordering system based on the values of notations. This issue is similar to the question of whether skos:hasTopConcept is needed because the top concept should be able to be traced all the way from the most specific concepts to the top concept based on the skos:broader property. But it is important to point out the difference: tracing to the top concept through skos:broader is based on the explicit expression of the relationships between and among concepts. The orders of sibling classes or concepts, however, are never expressed in SKOS.

To some degree, when order is connected to hierarchy, this can be reflected by extensions to SKOS. The DDC for example has two parallel hierarchies, one expressed by length of notation, the other by other structural devices (notes, etc.). This can currently be handled by extending skos:narrower:

```
skosclass:narrowerStructural rdfs:subPropertyOf skos:narrower .
skosclass:broaderStructural rdfs:subPropertyOf skos:broader ;
    owl:inverseOf skosclass:narrowerStructural .
```

Order that is not connected to hierarchy, however, becomes more difficult to express, especially in an RDF-based format. The abstract syntax for RDF statements is order-agnostic (*RDF: Concepts and Abstract Syntax*, 2004). Only at the level of certain serializations are sequences again introduced (addressable through XPath's axes in XML documents, for example).

SKOS provides for ordered collections that can be used in conjunction with skos:memberList (essentially using an construct already available in RDF) to express specific sequencing of concepts, e.g., to accommodate the main schedules. Since each concept has a unique URI in SKOS, gathering concepts into member lists is a straightforward process. But this simple sequencing of main schedules addresses only a subset of the issues regarding coordination of concepts. A combination of nested ordered and unordered collections, use of expressive notation

https://doi.org/10.23106/dcmi.952109518

that allows reliable sequencing, and other means discussed above might be necessary to adequately address this issue, necessarily accepting the implications of transferring semantic characteristics to inherently extra-semantic elements like skos:Collection.

3. Conclusion and further steps

After many unsuccessful attempts to make data exchange standards useful for controlled vocabularies "in the wild," SKOS promises to be here to stay. The reasons for this are manifold. Its "lowest common denominator" approach has streamlined the final specification into a core system that can be used as a jumping-off point in many different directions. The progression through the various maturity levels of the W3C ratification process has resulted in a rigorous vetting by stakeholders. By targeting terminologies that, from the Semantic Web point of view, appear to be legacy data, SKOS might help to bridge the gap between the primarily technology-driven vision of the Semantic Web community and the strict but somewhat insular practices of authority control developed and maintained by the library community, turning the tools that support these practices into machine-accessible sets of Linked Data.

The paper presents a selected set of issues in order to discuss the difficulties of expressing classification schemes without sacrificing a large degree of their semantic richness. SKOS extensions may solve some of the major problems. Future plans of the authors include testing and finalizing the extensions suitable for classification systems when SKOS is chosen as one of the encoding formats. The authors are also comparing and experimenting with both SKOS and OWL to ascertain which might better support different fundamental structures represented by thesauri and classification systems.

In addition to the attempt to influence the development of SKOS, there are various probabilities and usefulness of different strategic options—thanks to our anonymous reviewer who pointed these out—such as: (a) developing workarounds, hacks and multiple extensions to SKOS, (b) opting for a specific SKOS variant for KOS that are very different from thesauri; (c) trying different combinations of SKOS and OWL, (d) using OWL instead of SKOS; and/or (e) developing rich local semantics and representations. Considering these options, it might be worthwhile to emphasize that implementers should be clear about their specific use cases. SKOS plus extensions might not be able to represent any complex classification system completely, but may be instrumental for surfacing some of the most valuable assets of a system in an interoperable way that reveals other, emergent qualities.

To continue exploring these issues and options, the authors will have another attempt to bring together evidence of using OWL to resolve major issues related to classification systems and discuss the differences of SKOS and OWL that mainly support two different kinds of models underlying thesauri and classification systems. The results will be presented at the 2010 ISKO (International Society for Knowledge Organization) conference.

References

Berners-Lee, T. (2000, December 6). RDF and the Semantic Web. Presented at the XML 2000, Washington, DC. Retrieved July 28, 2009, from http://www.w3.org/2000/Talks/1206-xml2k-tbl.

Berners-Lee, T. (2007). Linked Data - Design Issues. Retrieved May 7, 2009, from http://www.w3.org/DesignIssues/LinkedData.html.

David Beckett, & Tim Berners-Lee. (2008). Turtle - Terse RDF Triple Language. Team Submission, W3C. Retrieved July 27, 2009, from http://www.w3.org/TeamSubmission/turtle/.

Iyer, H. (1995). Classificatory structures: concepts, relations and representation. Frankfurt am Main: Indeks Verlag.

OWL Web Ontology Language Guide. (2004). Recommendation, W3C. Retrieved August 3, 2009, from http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

Resource Description Framework (RDF): Concepts and Abstract Syntax. (2004). Recommendation, W3C. Retrieved August 3, 2009, from http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

https://doi.org/10.23106/dcmi.952109518

SKOS Simple Knowledge Organization System Primer. (2009). Working Draft, W3C. Retrieved May 7, 2009, from http://www.w3.org/TR/2009/WD-skos-primer-20090317/.

SKOS Simple Knowledge Organization System Reference. (2009). Candidate Recommendation, W3C. Retrieved May 6, 2009, from http://www.w3.org/TR/2009/CR-skos-reference-20090317/.

SPARQL Query Language for RDF. (2008). Recommendation, W3C. Retrieved July 27, 2009, from http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

Svenonius, E. (2000). The intellectual foundation of information organization. Cambridge Mass.: MIT Press.

Tennis, J. T., & Sutton, S. A. (2008). Extending the simple knowledge organization system for concept management in vocabulary development applications. Journal of the American Society for Information Science and Technology, 59(1), 25-37.

https://doi.org/10.23106/dcmi.952109518