SKOS: Requirements for Standardization

Alistair Miles
CCLRC Rutherford Appleton Laboratory
Tel: +44 (0) 1235 445440
Fax: +44 (0) 1235 445945
a.j.miles@rl.ac.uk

This paper poses three questions regarding the planned development of the Simple Knowledge Organisation System (SKOS) towards W3C Recommendation status. Firstly, what is the fundamental purpose and therefore scope of SKOS? Secondly, which key software components depend on SKOS, and how do they interact? Thirdly, what is the wider technological and social context in which SKOS is likely to be applied and how might this influence design goals? Some tentative conclusions are drawn and in particular it is suggested that the scope of SKOS be restricted to the formal representation of controlled structured vocabularies intended for use within retrieval applications. However, the main purpose of this paper is to articulate the assumptions that have motivated the design of SKOS, so that these may be reviewed prior to a rigorous standardization initiative.

Keywords:
SKOS, Semantic Web, RDF, Metadata, Information Retrieval, Controlled Vocabularies, Dublin Core.

1. Introduction

The Simple Knowledge Organisation System (SKOS) (1, 2) is a formal language for representing controlled structured vocabularies such as thesauri or classification schemes. This paper assumes that the reader already has a working knowledge of SKOS – see (1) or (18) for an introduction.

Recently, a Semantic Web Deployment Working Group (SWDWG) (19) has been chartered within the remit of the W3C's Semantic Web Activity. The SWDWG will be responsible for developing a W3C Recommendation based on current W3C Working Drafts for SKOS (1, 2), within a timescale of 18 months from September/October 2006.

In preparation for the W3C Recommendation Track process, this paper explores assumptions that have motivated the design of SKOS to date, so that these assumptions may be subject to critical review. The goal is to ensure that the motivational basis for the standardization of SKOS is sound, so that standardization may proceed to completion within the given timeframe and with the widest possible consensus.

This paper is structured around three questions:

1. What is the scope and purpose of SKOS?
2. What software components depend on SKOS, and how do they interact?
3. What is the wider social and technological context in which SKOS is likely to be applied?

Section 2 discusses the scope of SKOS, with a view to defining an unequivocal set of requirements. The section begins by describing a generic workflow involving the use of controlled structured vocabularies for retrieval, whereby a vocabulary is created, used to manually "index" a collection of items, and then used to "retrieve" items from the collection. It is suggested that this workflow be taken as a common framework for a set of use cases to be articulated at the outset of the standardization process. The formal requirements, and therefore scope, for a SKOS W3C Recommendation could be derived from an analysis of these use cases. Section 2 gives further suggestions as to how this analysis might be structured.

Section 3 focuses on assumptions regarding the software architecture within which SKOS is primarily to be applied. The practical goal of SKOS is to support the interoperation of software systems via a common data language. Section 3 therefore articulates assumptions regarding which software components are relevant and what their capabilities are. Three "key" software components are identified, referred to generically as: a "Vocabulary Development Application", an "Indexing Application" and a "Retrieval Application"; one associated with each of the steps in the generic workflow described in section 2. A functional specification is sketched for each of these components, and assumptions regarding the ways in which these components interact are discussed.

Section 4 analyses the wider technological and social context within which SKOS is likely to be applied. It is anticipated that a controlled vocabulary is unlikely to be applied as a complete retrieval solution, but as part of an integrated suite of solutions. Understanding this application context is a prerequisite to understanding trends in the use of controlled vocabularies and therefore trends in supporting software systems. Controlled vocabularies are costly to create and maintain and other technologies, such as text retrieval and social software, can both compete with and complement a controlled vocabulary solution. There is therefore a strong pressure to minimize the cost and maximize the utility associated with controlled vocabularies and to integrate functionalities wherever practical. Section 4 discusses this application context, and suggests ways in which cost can be reduced and utility increased, especially where this may have a bearing on the design of SKOS.

2. Defining the scope of SKOS

The design of SKOS has assumed that the principal motivating use cases all follow the generic workflow given below:

1. A controlled (structured) vocabulary is created.
2. An "index" over a collection of items is manually created using the controlled vocabulary.
3. The index and the vocabulary are used to "retrieve" items from the collection.

By "index" I mean a relation between items in a collection and conceptual units in a controlled vocabulary.

I suggested that the purpose of SKOS be broadly defined as supporting the use of controlled structured vocabularies for retrieval (not necessarily of documents). Furthermore, I suggest that the requirements for SKOS be defined by first articulating a set of concrete use cases that follow the generic workflow given above, then by analyzing these use cases and classifying them according to vocabulary type, index type and retrieval functionality.

This analysis should produce a set of vocabulary types (or structural features if it proves impractical to define distinct types), a set of index types (or structural features), and an associated set of retrieval functionalities. These three basic sets could be taken to define the formal requirements for SKOS, by stating that it must be possible to represent all the vocabulary features given in the first set, all the index features given in the second, and it must be possible to use those representations to implement the retrieval functionalities given by the third.

Each use case would therefore begin with an abstract description of the structural features of a controlled vocabulary. The use case would continue by describing in abstract the structure of the index, and conclude with an abstract description of the retrieval functionalities that are to be implemented. When analyzing retrieval functionalities, it may prove useful if each use case provides an independent description of the types of query that are supported and of the ways in which those queries are evaluated. This is because e.g. two systems might have different query capabilities, but might use the same underlying strategy to evaluate those queries.

Note that, in order to facilitate a comparative analysis of the use cases, it is important that each use case does not confuse the underlying structural features of a controlled vocabulary, with the ways in which that vocabulary might be presented via a user interface, or in print or other media. Similarly, it is important that retrieval functionalities should be described independently of the way in which those functionalities might be realized via a user interface. E.g. two applications might support the same types of query, and employ the same query evaluation strategy, but might have completely different interfaces for building queries, and might present the results in different ways.

Nevertheless, the presentation of a controlled vocabulary to a user is of course a central part of all associated information systems and therefore the types of presentation that must be supported should also be described for each use case.

Finally, each use case should articulate the application context within which the controlled vocabulary systems are to be deployed. This might include complementary solutions for achieving a suite of retrieval functionalities, such as text retrieval and/or social software, with which the controlled vocabulary systems must be integrated.

I sketch a hypothetical use case below, to illustrate some of the suggestions made above.

2.1 Example use case

Introduction: A web-based search portal is to provide online access to a large collection of reports (~1,000,000 items). Each report has semi-structured text content (i.e. headings, paragraphs etc.), and has metadata describing the subject(s) of the work using a controlled structured vocabulary.

Vocabulary Features: The controlled vocabulary used in subject metadata is large (~40,000 conceptual units). The vocabulary has the following features:

- Each conceptual unit has an identifier, which is a string of Unicode characters.
- Each conceptual unit has at least one lexical representation (i.e. label), which is a string of Unicode characters. If there are several labels, one of them is preferred. Only English language labels are given.
- Each conceptual unit has a definition in English, which is a string of unstructured text, and may have any number of other similar annotations of various types (all in English).
- The conceptual units are divided into two groups. The first group comprises the main body of the vocabulary. The second smaller group contains "qualifiers", which are conceptual units that may be “coordinated” with main body concepts to create more specific meanings.
- The conceptual units of the main body are linked via generalization ("broader" / "narrower") and via association ("see also") relationships. The generalization links define a poly-hierarchy – i.e. a conceptual unit may have more than one broader link.
- Each main body conceptual unit is linked to a list of "allowed qualifiers", i.e. those qualifiers with which it is sensible to be coordinated.

Index Features: Each report may have one or more allocations to describe the subject(s) of the work. Each allocation consists of a reference to a main body conceptual unit, or a reference to a main body conceptual unit coordinated with a reference to a qualifier.

Retrieval Functionalities: The search portal supports composite queries, i.e. queries that have one or more components. Each component of a query may be a reference to a main body conceptual unit, or may be a reference to a main body concept coordinated with a reference to a qualifier. Queries are evaluated against the index, and result sets are expanded via the generalization and association links in the vocabulary, according to an algorithm that assumes that relevance degrades linearly as links are followed away from the components of the query.

Application Context: The search portal incorporates text retrieval functionality, operating over the semi-structured text content of the reports. Also, the portal allows users to "bookmark" interesting reports, and to share bookmarks with other users.

3. Anticipated Software Architecture

A functionally distinct software component can be defined for each of the three tasks in the workflow described in the previous section (whereby a vocabulary is created, used to index a collection of items, and then used to retrieve items via the index). I refer to these three components generically as the "Vocabulary Development Application" (VDA), the "Indexing Application" (IA) and the "Retrieval Application" (RA) respectively.

A fundamental assumption in the design of SKOS has been that its ultimate purpose is to support the interoperation of these three "key" software components, via a common data language. I therefore sketch a functional specification for each of these components below, as assumed in the design of SKOS. These specifications, if reasonable, provide a list of the functionalities that SKOS must enable, or at least must not impede. Note also that these specifications include assumptions about trends in the evolving development of software systems – i.e. they are "ideal" and may not correspond to any current implementations.

A standard is irrelevant without implementation, and by articulating these particular assumptions, it is hoped that the current and near-future requirements for software systems associated with the application of controlled vocabularies for retrieval may be clearly established.

Note that a major assumption regarding these components is that they interact (i.e. exchange data) via the infrastructure of the Semantic Web. This interaction model assumes that each component publishes the data it produces in the Semantic Web, e.g. the VDA publishes the vocabulary, and the IA publishes the index, according to current best practice for the publication of RDF data (14). The RA then accesses the vocabulary and the index data via Semantic Web interaction protocols, i.e. via HTTP and SPARQL. This interaction model has been assumed because it is anticipated that SKOS data will commonly be deployed in a context where data is being linked to and/or merged with data from other sources, e.g. two or more indexes refer to the same vocabulary and are being merged. The infrastructure of the Semantic Web (i.e. URI, HTTP, RDF, SPARQL) provides particular support for this kind of scenario.

3.1 Ideal Specification of a Controlled Vocabulary Development Application

The Vocabulary Development Application is a development environment for controlled structured vocabularies. It allows one or more people to collaboratively create and edit a vocabulary. This component supports a continuum of collaboration models, from strict editorial control to "anarchy of the masses" and appropriate change management procedures are built in to its workflow.

Current implementations tend to specialize on particular type of vocabulary (e.g. thesauri conforming to ISO 2788:1986 (4)) however a complete implementation will allow the construction of a controlled structured vocabulary according to a custom profile of features from thesauri, classification schemes, subject heading systems and/or taxonomies.

Simple multilingualism (e.g. via multilingual labels (1) and annotations) is supported. Custom extensions to the basic vocabulary structure (for example, custom semantic relations) are also supported. Finally, publication of a vocabulary in the Semantic Web is handled transparently.

3.2 Ideal Specification of a (Controlled Vocabulary) Indexing Application

The Indexing Application is a development environment for creating and maintaining an index over a collection of items using a controlled structured vocabulary. It allows one or more people to collaboratively develop an index. The index is typically subject-based; however, indexes over multiple metadata fields are supported, and therefore this component supports the creation and/or import of a custom metadata application profile.

The work of the indexer is automated as much as possible, by whatever means available (e.g. via statistical comparison of the text content of indexed items with text labels and annotations in the controlled vocabulary and/or via machine learning algorithms operating on the index-so-far as the training set).

Current implementations tend to focus on a particular style of index assuming a particular type of controlled vocabulary (e.g. subject classification with a classification scheme), however, a complete implementation supports construction of an index according to a custom profile of features, e.g. both primary and secondary subject allocations, with support for coordination.

This component is able to interact with the indexing vocabulary via the Semantic Web and is also able to handle vocabulary changes in a sensible way. Finally, publication of an index in the Semantic Web is handled transparently.

3.3 Ideal Specification of a (Controlled Vocabulary) Retrieval Application

The Retrieval Application allows users to interact with one or more indexes over one or more collections using one or more controlled structured vocabularies. Current implementations of this component tend to encapsulate an index over a single collection; however, future implementations are able to take advantage of the ability to merge Semantic Web data, and thereby encapsulate a combined index over many collections.

This component is able to calculate relevance metrics in order to improve search precision by effective ranking, and is able to expand result sets in order to improve search recall. These functions require exploitation of both vocabulary structure and latent index structure (i.e. both paradigmatic relationships between conceptual units asserted in the vocabulary, and syntagmatic relationships between conceptual units discovered from an analysis of the available indexes (16)).

Where retrieved items are resources that have text content, this component integrates transparently with other components providing text retrieval functionality.

4. Social and Technological Context for the Application of Controlled Vocabularies

Creating and maintaining a controlled vocabulary and using a controlled vocabulary to establish an index over a collection of items, are costly endeavours, primarily because the extent to which they can be automated is very limited. Most of the work must be done by hand, and to ensure quality and consistency, by people with specific expertise and training. Furthermore, a retrieval service based on this process does not provide any real value until the vocabulary is relatively complete and stable, and the index has been created and quality-controlled. I.e. bootstrapping a service of this kind requires a significant initial investment of both time and money. In addition, as end user needs evolve, so the vocabulary and the index must evolve if the service is to remain relevant – and thus a significant ongoing cost may be involved.

Two fundamental questions therefore are: (i) under what circumstances is it "profitable" to provide a retrieval service using a controlled structured vocabulary (i.e. when does the "value" outweigh the cost), and (ii) what strategies can be employed to reduce the initial and ongoing costs? These questions are relevant because they indicate (i) when SKOS is likely to be applied, and (ii) how the design of SKOS can support the minimization of cost and maximization of value.

This section begins with a brief overview of other strategies that may directly compete with and/or complement retrieval solutions based on a controlled vocabulary, with a view to developing an understanding of when a controlled vocabulary is likely to be “profitable”. This section continues by discussing ways in which the “profitability” of a controlled vocabulary may be maximized, and how this might impact on the design of SKOS.

4.1 Competing and/or Complementary Solutions for Retrieval Services

The following discussion assumes that the end user of a retrieval service has two fundamental requirements: (a) to be able to *locate* an item in a collection, where the user has some prior knowledge of the item, and (b) to be able to *discover* items in a collection that are relevant to the user's interest, of which the user has no prior knowledge.

If items have some text content, then basic location and discovery services can be achieved by well understood text retrieval methods, which can be fully automated. The performance of these services can be improved if the content is semi-structured in a consistent way, for example if the items are documents that contain headers at multiple levels (e.g. as in HTML).

Basic descriptive metadata, such as title, description, author etc. can also be used to achieve basic location and discovery services. In the absence of any text content, this is probably the cheapest way of implementing some sort of retrieval service. In the presence of unstructured or semi-structured text content, this metadata may add value to services supported by text retrieval (although it may not, cf. the Web).

If items refer to other items in the collection (e.g. via hyperlinks, or via bibliographic citations), then the topology of the directed graph derived from these references can be exploited to significantly improve the performance of retrieval services. For example, Google’s “pagerank” algorithm (17) ranks web sites based on the number of inward links, which are themselves weighted by the rank of the referring site. Such strategies depend on the assumption that the number of inward references is indicative of quality, importance and/or relevance.

If the behaviour of the end users of a retrieval service can be captured and correlated, then an analysis of this behaviour can be used to provide retrieval services, and/or to improve services achieved by other means. A simple example is Amazon's referral service (i.e. "people who bought X also bought Y and Z"). A more sophisticated example is provided by social bookmarking websites such as del.icio.us. Because users only bookmark web sites they have visited and found useful, the number of times a site is bookmarked provides a quality or importance metric. Also, because users may discover other users with similar interests via the bookmarks they have in common and/or via the "tags" they both use, a social network is established. Users can exploit this social network to discover web sites in their domain of interest.

These examples illustrate how a retrieval service can mediate social interaction between its users. This in turn allows knowledge of the collection to propagate through a user community. Moreover, by allowing users to express interest in specific items, a network of interest emerges, the topology of which can be exploited to improve the performance of existing services (as e.g. pagerank improves the performance of Google’s text retrieval functionality) and/or provide additional services (as e.g. Amazon's referrer service).

The bottom line for controlled vocabularies is that text retrieval and/or facilitated social interaction can underpin fully automated services, which therefore incur minimal ongoing costs, and low initial investment at least in terms of human effort. Therefore, if a controlled vocabulary is to be "profitable", the value it adds *over and above* that provided by other means must outweigh the costs of manual creation and curation of vocabularies and indexes. Whether this is the case for any given collection or group of collections will depend on many factors, in particular the nature and scope of the items, the dynamics of the collection and of the user community, and the specific needs of the end users.

4.2 Maximizing the Profitability of Controlled Vocabularies

How can the costs associated with controlled vocabularies be reduced?

One way is to decentralize responsibility for creating and maintaining the vocabulary. I.e. allow an open-ended community of developers to work on the vocabulary, preferably where the developers are also the end users of the vocabulary. The costs of creation and maintenance are distributed and shared. However, quality and consistency may suffer. Also, applications may have to cope with continuous change.

There is a continuum of collaboration models between strict editorial control and "anarchy of the masses" and being able to explore this continuum may be a crucial factor in finding the ideal balance between cost and quality for any given application. An intermediate collaboration model might involve an editorial team delegating responsibility for parts of a vocabulary to different groups, and holding special privileges to ensure consistency. Under a totally decentralized collaboration, a vocabulary will typically be under continuous change. Under strict editorial control, a vocabulary is typically under a discrete change management model, where versions are periodically released. An intermediate collaboration model might employ a change management process that has both continuous and discrete aspects.

Because a controlled vocabulary represents a large investment, there is significant motivation for maintaining the vocabulary over time. However, because the lifetime of a controlled vocabulary may be of the order of tens of years, retrieval applications must respond to change if they are to continue to provide consistent and relevant results. A representation framework such as SKOS must therefore provide declarative support for describing change in controlled structured vocabularies, in order to support adaptation in retrieval applications. Furthermore, this declarative support must accommodate both discrete and continuous management models.

5. Conclusions

This paper has sought to articulate the main assumptions that have motivated the design of SKOS to date, with a view to ensuring that only those assumptions which are sound are carried into the planned development of SKOS towards W3C Recommendation status.

A first critical assumption has been that the ultimate purpose of SKOS is in support of the use of controlled structured vocabularies for retrieval. Specifically, the motivating use cases for SKOS follow a common workflow, whereby a controlled vocabulary is created, is used to construct an "index" over a collection of items and is then used with the index to "retrieve" items from the collection.

This paper has suggested that the process of defining formal requirements for SKOS begins by articulating a core set of use cases, which all follow the workflow given above. It is furthermore suggested that the formal requirements be obtained via an analysis of these use cases, where each use case is classified according to (a) the structural features of the vocabulary, (b) the structure of the index and (c) the retrieval functionalities provided. This analysis will provide a set of vocabulary features and a set of index features, for which it must be possible to use SKOS to represent, and a set of retrieval functionalities, for which it must be possible to use the SKOS representations of the vocabulary and the index to implement.

A second critical assumption has been that the practical purpose of SKOS is to enable the interoperability of three functionally distinct "key" software components, which interact via the architecture of the Semantic Web. These components, generically, are referred to as a "Vocabulary Development Application", an "Indexing Application" and a "Retrieval Application". SKOS has been intended to provide an application layer data language whereby the data published via the first and second of these components may be interpreted by the third.

This paper has sketched a functional specification for each of these key components, with a viewing to determining the practical needs of the users at each of the stages in the workflow, and with a view to anticipating current trends in the development of software applications associated with controlled vocabularies.

A third critical assumption has been that, although there is renewed interest in the use of controlled vocabularies to provide high precision retrieval services, there are alternative technological frameworks which in some cases complement, and in other cases compete directly with the use of controlled vocabularies. The utilitarian purpose of SKOS is, therefore, to support the minimization of cost and maximization of value associated with the application of controlled vocabularies. In practical terms, this means at least being aware of, if not directly supporting, a range of models for collaborative development and change management. This also means enabling maximal interoperability between similar vocabulary types, e.g. between thesauri and classification schemes, in addition to providing an extensibility mechanism to ensure that specific needs can be met without compromising interoperability, or where that is impossible, with a graceful degradation of interoperability.

Whereas previously a solution based on a controlled vocabulary may have been deployed as a complete solution for retrieval, it is anticipated that controlled vocabularies are likely to be applied as part of a suite of solutions, in parallel with services based on text retrieval, on mediated social interaction and/or on analysis of reference and/or social networks. The development of a standardized representation language for controlled vocabularies must at least be aware of this evolving application context.

6. References

1. SKOS Core Guide, Alistair Miles and Dan Brickley, World Wide Web Consortium, W3C Working Draft, November 2005.

2. SKOS Core Vocabulary Specification, Alistair Miles and Dan Brickley, World Wide Web Consortium, W3C Working Draft, November 2005.

3. An RDF Schema for Thesauri (SKOS-Core 1.0 Guide), Alistair Miles, Nikki Rogers and Dave Beckett, SWAD-Europe Project, deliverable (8.1), 2004.

4. Documentation - Guidelines for the establishment and development of monolingual thesauri, ISO, International Organization for Standardization, ISO (2788), 1986.

5. RDF Encoding of Classification Schemes, Alistair Miles, SWAD-Europe Project, deliverable (8.5), 2004.

6. Migrating Thesauri to the Semantic Web, Alistair Miles, Nikki Rogers and Dave Beckett, SWAD-Europe Project, deliverable (8.8), 2004.

7. RDF Vocabulary Description Language 1.0: RDF Schema, Dan Brickley and R. V. Guha, World Wide Web Consortium, W3C Recommendation, February 2004.

8. RDF Semantics, Patrick Hayes, World Wide Web Consortium, W3C Recommendation, February 2004.

9. W3C Glossary Project.

10. Computer applications in terminology - Terminological markup framework, ISO, International Organization for Standardization, ISO (16642), 2003.

11. Computer applications in terminology - Data categories, ISO, International Organization for Standardization, ISO (12620), 1999.

12. SPARQL Query Language for RDF, Eric Prud'hommeaux and Andy Seaborne, World Wide Web Consortium, W3C Candidate Recommendation, April 2006.

13. SPARQL Protocol for RDF, Kendall Grant Clark, World Wide Web Consortium, W3C Candidate Recommendation, April 2006.

14. Best Practice Recipes for Publishing RDF Vocabularies, Alistair Miles, Thomas Baker, Ralph Swick, World Wide Web Consortium, W3C Working Draft 14 March 2006.

15. DCMI Metadata Terms, DCMI Usage Board, Dublin Core Metadata Initiative, DCMI Recommendation, June 2005.

16. Structured vocabularies for information retrieval - Guide - Part 2: Thesauri, BSI, British Standards Institution, BS (8723-2), 2005.

17. The Anatomy of a Large-Scale Hypertextual Web Search Engine in Proc. 7th International World Wide Web Conference, Sergey Brin and Lawrence Page, April 1998.

18. SKOS Core: Simple Knowledge Organisation for the Web in Proc. International Conference on Dublin Core and Metadata Applications, Madrid, Spain, A. Miles, B. Matthews, M. D. Wilson and D. Brickley, September 2005.

19. W3C Semantic Web Deployment Working Group (SWDWG) Charter.
Available at http://www.w3.org/2006/07/swdwg-charter