Until recently, tools for managing controlled vocabularies, taxonomies, and thesauri all stored their data using proprietary formats that made the data difficult to integrate with external systems. The W3C SKOS standard defines a portable, flexible controlled vocabulary format that is increasingly popular, with the added benefit of a good entry-level step toward the use of Semantic Web technology.

Whether you manage a taxonomy to integrate business processes in an enterprise, to manage keywords assigned to content for more intelligent retrieval, or to manage the menus of a large web-based retail site, you might find that your taxonomy management tool stores data in a proprietary binary format that doesn't migrate well to other tools. A standards-based way to represent this data can help you integrate vocabulary data from multiple sources while reducing your dependence on proprietary tools.

Controlled vocabularies, taxonomies, and thesauri: What's the difference?

A **controlled vocabulary** is a list of terms that define the potential values for something—for example, the possible subjects of a set of news stories or the official two-letter abbreviations of the states of the United States. A **taxonomy** is a controlled vocabulary arranged in a hierarchy to show relationships between terms. The possible subjects of a set of news stories is most likely this kind of controlled vocabulary, with "Acquisition" and "Executive hiring" as children of the hierarchy's "Business news" node.

These relationships are metadata that indicate, for example, that a story about an executive being hired is a type of business news story or that a dachshund in an animal taxonomy is a kind of dog. When a taxonomy-aware image search engine returns a picture tagged "dachshund" to someone searching for "dog" pictures, it takes advantage of this metadata to help the searcher get greater value from the image collection.

A **thesaurus** is typically a taxonomy with additional metadata about each term such as alternative terms (for example, "mutt" for "dog") and pointers to related terms that might or might not be in the same hierarchy (for example, "doghouse" for "dog"). People who specialize in the creation and maintenance of thesauri are usually known as taxonomists.
The Simple Knowledge Organization System (SKOS) is a W3C standard that builds on the W3C’s RDF, RDFS, and OWL specifications to provide a standard model for representing controlled vocabularies. You can use SKOS for flat lists and also for more structured controlled vocabularies with additional metadata such as taxonomies and thesauri.

Because SKOS is defined using the RDF model, it's easy to read and create data in an XML format. Growing tool support for SKOS means that using it requires no knowledge of the related W3C standards, but the more you know, the more you can take advantage of the extensibility of SKOS to include customized metadata in your vocabularies that might not be part of the SKOS standard.

As organizations ranging from The New York Times to NASA to the UN Food and Agriculture Organization make their subject listings available in SKOS, this standard also makes it easier to reuse well-known vocabularies and to create connections between your content and other content that uses the same vocabularies.

Terms versus concepts and labels

Vocabulary management systems have always been structured to manage terms, along with relationships between terms and other metadata. SKOS takes a higher-level view of what you manage, which makes internationalization much easier. For example, an older system might store the term "dog" with a broader term of "mammal" and narrower terms of "dachshund" or "bulldog." The term "mutt" would be a separate term, and "dog" would have what taxonomists call a use-for relationship to "mutt"—if someone assigning keywords to photographs wants to assign the word "mutt" to a picture of Lassie, the vocabulary application would direct him to use the word "dog" instead. The term "perro" could have a relationship "Spanish" to the term "dog," and "chien" could have the relationship "French" to it, but a Spanish user wondering about the French term for "perro" might not be able to look this up without knowing that they're connected by their relationship to the English term.

Another disadvantage of this arrangement is that the terms "mutt" and "perro" are as separate from "dog" as the term "cat" or "gato" (a Spanish term). Even though "mutt," "dog," and "perro" refer to the same thing, their relationships must be explicitly specified. Figure 1 displays these relationships in a diagram; solid-line arrows represent a "broader than" relationship (mammal to cat and dog; dog to bulldog and dachshund), and dotted-line arrows are labeled for the Spanish ("perro") or French ("chien") equivalents for "dog," alternate terms in Spanish ("chucho") and English ("mutt") for "dog," plus the Spanish ("gato") for "cat."
Figure 1. Sample label relationships in a pre-SKOS taxonomy

With SKOS, you manage concepts that have different kinds of labels, and each label might have a language associated with it. The most important label is the preferred label, and SKOS allows each concept to have only one of these in each language. A single concept could have an English preferred label of "dog," a Spanish preferred label of "perro," and a French preferred label of "chien."

Another kind of label is the alternative label, which SKOS-based software might use to represent labels that are being tracked but not recommended. For example, the concept with an English preferred label of "dog" might have an English alternative label of "mutt" and a Spanish alternative label of "chucho." Instead of being separate terms that must have their relationships explicitly typed, "dog," "perro," "chien," "mutt," and "chucho" all refer to the same concept, providing different information about that concept depending on the needs of each application. Figure 2 illustrates the information from Figure 1 rearranged as SKOS concepts, with fewer arrows and clearer relationships between the terms. (As with the earlier figure, solid-line arrows represent a "broader than" relationship.) The actual identifiers for each concept, which might be hidden under the covers by a vocabulary management application, are URIs. (View a text-only version of Figure 2.)
When you compare the two diagrams, you can see that in Figure 1, "perro" and "mutt" were just additional terms that "dog" pointed to, "bulldog" and "dachshund," but in Figure 2 you can see that "perro" and "mutt" refer to the same concept while "bulldog" and "dachshund" are different concepts.

Concepts can have many kinds of relationships in SKOS besides "broader than." The concept with an English preferred label of "dog" might have a "related" relationship with a "doghouse" concept in a different taxonomy. Because SKOS uses unique URIs as concept identifiers instead of the labels themselves, you can define relationships between a given concept and any concept in any accessible SKOS vocabulary in the world, even if it's maintained by NASA or The New York Times.

The UN Food and Agriculture Organization's AGROVOC thesaurus for food-related domains such as fishing and farming must serve a truly international audience. A single AGROVOC concept can have preferred labels in over a dozen languages and even more alternative labels because there is no limit to the number of alternative labels you can specify for a given concept from each language. SKOS uses concepts with label properties to make multi-lingual tracking of terms much easier than one of the older, term-based approaches to organizing thesaurus data would, and this in turn makes communication between people from different cultures about food issues much easier.

More metadata

Along with the preferred and alternative labels and relationships between concepts described above, SKOS lets you store a term's definition, scope notes, history notes, and a variety of other properties about each concept. Because SKOS is defined using the W3C's OWL standard for specifying ontologies, it's very easy to define and use additional properties that are specific to your industry or business to the concepts in your vocabularies.

These properties can come from other data and metadata standards, such as the Dublin Core vocabulary, the Market Data Definition Language developed for the financial industry, or the
Metadata Object Description Schema developed by the Library of Congress. They can also be properties that are specific to your company's system and that no one else uses because they're part of the added value for how you manage your information. For example, a pharmaceutical company might define a new "requires" relationship in an animal taxonomy to point to concepts in another taxonomy's data about veterinary vaccines.

SKOS-based tools for editing and managing your vocabularies should understand that extensibility is part of this standard. Additional properties from outside of the SKOS specification should be part of their interface as you work with that data, showing up on the forms and reports along with the standardized SKOS properties.

**More granular metadata: SKOS-XL**

Although the OWL language used to specify SKOS has certain crucial differences from object-oriented approaches to data modeling, it has one important thing in common: You define a data model by declaring classes, subclasses, and properties (or, to use the object-oriented term, attributes) of those classes. The SKOS ontology defines a Concept class, and preferred labels, alternative labels, and relationships to other concepts are modeled as properties of that class.

You can assign all the metadata you want to a given concept, but SKOS provides no way to assign metadata to a specific label. What if you want to store data that describes the source of the label "chucho," or when it was last edited, or who edited it?

To accommodate this situation, the W3C published the SKOS Extension for Labels (SKOS-XL) specification, in which the values for a concept's preferred, alternative, and other labels are not strings but members of a new Label class defined by the extension specification. Being instances of a class, these labels can have all the metadata you want to assign to them, which gives you a lot more flexibility.

**Easier metadata integration**

Earlier I mentioned that because SKOS uses unique URIs as concept identifiers, you can define a relationship between a given concept and any other SKOS-based concept whose URI ID you know, whether it's in the same taxonomy as a given concept or in a different taxonomy published on the web by a separate company. This ability is also great for a situation that falls between these two extremes: When different groups within the same enterprise have their own vocabularies to manage, integration of these vocabularies into a centrally managed single vocabulary can do more harm than good because vocabulary maintenance becomes more complex with the growing scale of data and the data must be revised to reach compromises between the needs of different groups. The marketing department and the repairs department might mean different things when they use the term "customer," and they might have good reasons for doing so; forcing them both to use the same definition can reduce the vocabulary's value for both of them.

With SKOS, you can define relationships between concepts from different vocabularies. Because of this, well-defined concept relationship metadata gives you the hooks to use vocabularies from different departments together without forcing you to revise and combine them all into a monolithic
single vocabulary that doesn't fully meet any group's needs. The relationships can be standard SKOS relationships such as "related" or "broader" (for example, you might say that the marketing department's concept of "customer" is broader than the repair department's), but again, you can define your own customized relationships as well.

**SKOS and the Semantic Web**

When becoming interested in semantic technology, many worry that before they build their first application, they must learn the RDF data model, the various syntaxes for expressing it, the SPARQL query language, and how to model data with RDF schema and OWL. When you use a SKOS-based vocabulary manager, you most likely fill out forms and use typical user interface widgets to manage your data with no need to learn the base W3C standards that underlie SKOS, but if you choose to learn a little about them, you can get more out of your data. For example, you can use the SPARQL query language to ask questions that might not be part of your vocabulary management package, and as mentioned above, you can define new properties and even classes to keep track of more customized metadata.

You can also connect your data to a wider variety of data out there, whether it uses the SKOS ontology or not. The ability of the RDF data model to connect independently created data is what makes the Semantic Web a web, and the ability to combine datasets is an important payoff of this ability. For example, by making their SKOS-based subject header index freely available on the web, The New York Times lets other publishers use these subject headers for their own content, giving those publishers connections to related New York Times articles. More importantly, for The New York Times, it drives more traffic to their articles tagged with those subject headers.

After you've added some properties to your SKOS data and run a few SPARQL queries against it, you can think about defining new ontologies apart from SKOS (or finding other existing standard ontologies besides SKOS to extend) and take greater and greater advantage of Semantic Web technologies.

**Tools**

Any RDF tool that can edit data guided by a particular ontology can load the SKOS OWL ontology and let you create SKOS concepts and populate their properties with the appropriate metadata. For management of vocabularies by staff with no RDF background, several tools are available:

- TopQuadrant's Enterprise Vocabulary Net (EVN) is a commercial web-based collaborative system built around the SKOS data model for the management of controlled vocabularies across an enterprise.
- PoolParty is a commercial thesaurus management and SKOS editor system that includes text mining and linked data capabilities.
- The SKOSed plug-in for the Protégé ontology editor lets you edit thesauri represented in SKOS. Both SKOSed and Protégé are open source.
- iQvoc is an open source tool for managing vocabularies that can import and export SKOS.
- TemaTres is an open source vocabulary manager that can output vocabulary data as SKOS files.
Import and export of SKOS by vocabulary management tools should eventually be as common as import and export of comma-separated values by spreadsheet programs. If you use a taxonomy management program that doesn't support the standard, let its makers know that you want to see it.

The RDF basis of SKOS also means that you can take advantage of RDF-aware application development tools and libraries to build SKOS editing systems yourself much more quickly than you can build a taxonomy management system where you had to define and implement all the data structures yourself.

**Starting small and scaling up**

If you have one or more large, complex controlled vocabularies to manage, converting it all to use a new format can be a big, expensive job. Converting a subset to SKOS as a pilot project can be much easier, and if you convert a few different subsets and then eventually connect them by defining the appropriate concept relationships across vocabulary boundaries, you start to see the benefit of SKOS in your own organization. With the growing support of both free and commercial software for the standard, SKOS is definitely worth further investigation by anyone who manages vocabularies and is interested in the benefits of standardization.
Related topics

- **Subject classification with DITA and SKOS** (Erik Hennum, Robert Anderson, and Colin Bird; developerWorks, October 2005): Learn how SKOS can help to organize DITA topics for better searching, integration, and other processing of your content.
- **Integrate disparate data sources with Semantic Web technology** (Bob DuCharme, developerWorks, Sep 2010): See how the RDF data model can ease the aggregation of local and remote data that isn’t necessarily in RDF.
- **SKOS Simple Knowledge Organization System Reference** (W3C Recommendation, August 2009): Learn the official details of SKOS from the W3C specification.
- **SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL) Namespace Document - HTML Variant** (W3C Recommendation, August 2009): Explore the W3C specification for the SKOS extension that allows more granular metadata.
- **SKOS Datasets** Explore publicly available SKOS datasets from a variety of providers on this W3C list.
- **Enterprise Vocabulary Net (EVN) by TopQuadrant**: Explore a web-based solution for simplified development and management of interconnected controlled vocabularies, taxonomies, thesauri, and ontologies.
- **SKOSed plug-in for the Protégé ontology editor**: Create and edit thesauri represented in the Simple Knowledge Organisation System (SKOS) with this plugin.
- **TemaTres**: Download TemaTres software for free at SourceForge.net and try this web application for management formal representations of knowledge, thesauri, taxonomies and multilingual vocabularies.
- **More articles by this author** (Bob DuCharme, developerWorks, November 2010-current): Read articles about semantics, DITA and other technologies.
- **XML area on developerWorks**: Get the resources you need to advance your skills in the XML arena. See the **XML technical library** for a wide range of technical articles and tips, tutorials, standards, and IBM Redbooks
- **IBM XML certification**: Find out how you can become an IBM-Certified Developer in XML and related technologies.
- **IBM product evaluation versions**: Get your hands on application development tools and middleware products.