SEMANTIC INTEROPERABILITY BETWEEN SKOS CONCEPT SCHEMES USING METADATA

SALVADOR SÁNCHEZ-ALONSO
ELENA GARCIA-BARRIOCANAL
University of Alcalá
Ctra. Barcelona km. 33,6 – Alcalá de Henares 28871, SPAIN
E-mail: {salvador.sanchez,elena.garciab}@uah.es

The SKOS (Simple Knowledge Organization System) Core is is a model for representing thesauri and similar types of knowledge organization systems as RDF graphs. Although it provides a basic framework for building concept schemes, it does not carry the strictly defined semantics of ontology languages and thus has a number of shortcomings to fully port existing schemes to the Semantic Web. This paper introduces a mapping of SKOS metadata to an ontology-based intermediate model which includes terms from upper ontologies, whose main aim is to foster the semantic interoperability of different concept schemes. It has been achieved through the introduction of a common ground for the definition of concepts, based on the use of shared definitions already included in widely-used upper ontologies. This effort makes use of *OpenCyc*, the open source version of *Cyc*, which is currently the world's largest and most complete general knowledge base.

1 Introduction

The SKOS Core [8] is an application of the Resource Description Framework (RDF) that allows expressing a concept scheme as an RDF graph by using a number of terms known as the *SKOS Core Vocabulary* [10]. Concept schemes, as defined by SKOS, are "thesauri, classification schemes, subject heading lists, taxonomies, terminologies, glossaries and other types of controlled vocabularies". Thus, the metadata elements in the SKOS vocabulary allow to represent the content and structure of concept schemes (particularly those that have a specific structure described by the SKOS Guide) with the aim of promoting their use by Semantic Web applications.

An example of the representation of a concept scheme in SKOS is the extract of the getty AAT (*Arts and Architecture Thesaurus of geographic names*, http://www.getty.edu/research/conducting_research/vocabularies/aat) shown in Table 1. The information and structure of this extract can be represented as an RDF graph as shown in Figure 1. This example illustrates how SKOS can be used to map a given thesaurus to RDF.

Using RDF graphs as a representation mechanism has a number of benefits, such as allowing data to be linked to other RDF data by Semantic Web applications, or providing serialization capabilities for concept schemes to be encoded as a series of characters according to a number of RDF syntaxes (RDF/XML, N3/Turtle or N-Triple). However, although this is an important

Table 1. An extract of the getty AAT concept scheme: the term *castles*.

Term	castles
Used for	buildings or groups of buildings intended primarily to serve as a fortified residence of a prince or nobleman
Broader terms	fortifications
Narrower terms	chatelets moated castles qasrs
Related terms	fortification elements

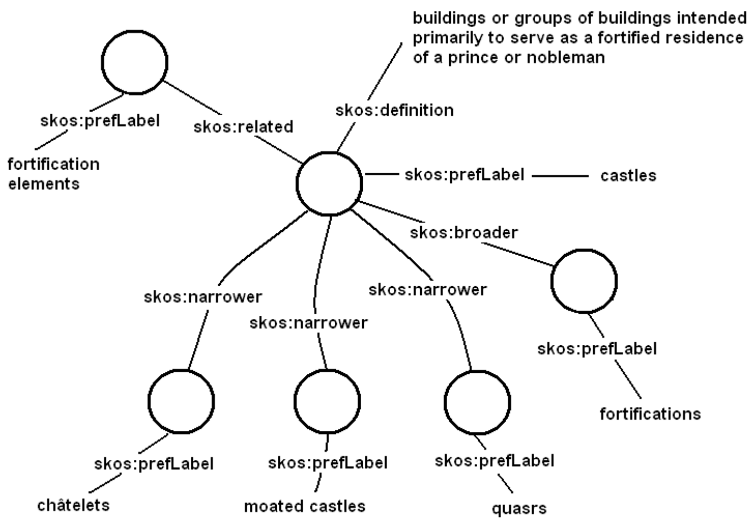

Figure 1. The same extract of the getty AAT expressed as an SKOS RDF graph.

step towards the use of a particular vocabulary by applications, two important shortcomings can be identified:

- First, the representation of a concept scheme as an RDF graph is not, by itself, enough to make it interoperable because the meaning of the terms in the SKOS vocabulary is not formally defined. Providing formal definitions for SKOS terms would both prevent the inherent ambiguities in the interpretation of some terms, and ease the shared use of SKOS-based schemes by including more precise definitions.
- Second, the existence of similar concepts in different schemes suggests the possibility of establishing mapping criteria to foster the interoperability

between them. Regarding this, the so-called *SKOS Mapping Vocabulary Specification* draft [9] is oriented to give support to mappings between concepts from different schemes, but is currently in a very early stage.

These two shortcomings can be summed up in one sentence: SKOS does not provide computational semantics. That is to say, the representation of a concept scheme as an RDF graph can not be used as the basis for performing automated tasks associated to the knowledge represented in the scheme. In fact, when some degree of automation is desired, the provision of a specific ground for the delegation of tasks to automated or semi-automated systems is necessary. Formalisms such as description logics [2] provide support for the explicit definition of terms and properties oriented to shared management, automated processing and reasoning based on specific inference mechanisms. The following sections will show how the use of ontologies can significantly improve the interoperability of concept schemes, as the inner description logics which provide introduce the necessary degree of formalization.

The rest of this paper is structured as follows. Section 2 provides a brief introduction to the field of ontologies, particularly those known as upper-ontologies, and the benefits of their use for the knowledge representation of concept schemes. Section 3 introduces two ontology-based proposals aimed at fostering semantic interoperability [4] between SKOS concept schemes. Finally, section 4 provides conclusions and some directions for further research.

2 The role of ontologies

As it has been remarked in the previous section, RDF representations are a big step towards permitting Semantic Web applications to use and manage concept schemes. However, a number of shortcomings were pointed out. This section deals with the problem of semantic interoperability between thesauri and introduces ontologies as a useful tool towards attaining it.

Most thesauri include terms that are also part of other thesauri, sometimes with the same meaning, sometimes with a very close meaning and sometimes with a meaning significantly different. For example, the term *back* in a anatomy thesaurus will be defined as "the rear part of the human body, especially from the neck to the end of the spine", while a sport glossary could include the same term with a slightly different meaning: "a position behind the front line of players". It is also feasible to think that the latter glossary could include a second definition of the term, to designate "a player in the back position". The example points out that any term in a SKOS scheme can have different meanings and consequently refer to different terms in a knowledge base.

One of the main difficulties in attaining interoperability (as a general feature) is the lack of explicit, shared definitions that allow to unambiguously refer to a term. To avoid this problem, thesauri should include formal definitions of all the terms and relations, which should be achieved by making use of a specific formal language (i.e. mathematical or logical). Definitions like these would be then referred to as "semantic definitions". Unfortunately, this is not the case of neither the terms in the SKOS vocabulary nor the terms in most thesauri, as they do not provide support for the so-called *semantic interoperability*. For the purpose of this work, semantic interoperability will be defined as "the use of explicit semantic descriptions to facilitate concept scheme integration with the main objective of fostering the (semi-)automated use of the information". Ontologies will be introduced as a tool to attain semantic interoperability in SKOS concept schemes.

In the field of philosophy, the term *ontology* is defined as the theory of objects and their ties. Therefore, the definition of a shared ontology for a given domain provides criteria for distinguishing different types of objects in the domain as well as their relations [3]. Outside philosophy, ontologies can be understood as conceptualizations that provide an appropriate context for the interpretation of concepts in a given domain. An often-cited definition of the term by Gruber [5] states that an ontology is "a specification of a conceptualization". In this sense, ontology engineering becomes of particular interest when applied to conceptual modeling.

The existence of ontology-based schemes in a domain of discourse is essential when some degree of automation is desired. The inner logics in the ontology allows automated systems to perform tasks according to the elements defined, which is the basis for applying the principles of Semantic Web in the domain of the ontology. However, creating a new ontology from scratch is a huge effort, as it would imply to define all the elements (terms and relations) needed before the terms in the current concept scheme can be explicitly defined and situated in the right place in the full hierarchy of concepts. To avoid defining time and time again all the concepts from which others derive, *upper ontologies*, large general knowledge bases that include definitions of concepts, relations, properties, constraints, and instances, as well as reasoning capabilities on these elements, can be used. They are limited to generic, high-level, abstract concepts, general enough to address a broad range of domains, not including concepts specific to given domains, or do not focusing on them. One of the major efforts in the field is *Opencyc* (http://www.opencyc.org), an upper ontology "for all of human consensus reality" which includes more than 47,000 concepts, 300,000 assertions about them, an inference engine, a browser for the knowledge base and other useful tools. It is the open source version of the larger *Cyc* knowledge base [6], a huge representation of the fundamentals of human knowledge.

After an in-depth study of the SKOS vocabulary, its extension with the aim of correcting the shortcomings identified in section 1 emerges. However, although such an extension would help both to avoid ambiguities and to enable inter-thesaurus semantic interoperability, the solution to these problems should be better focused as a non-invasive contribution. Non-invasive in the sense that the SKOS Core shouldn't be modified as a result of this work, but also, non-invasive in the sense that current SKOS schemes should not require modifications. The use of formal representations to provide the SKOS terms with computational semantics, as well as the introduction of an intermediate ontology-based model on top of the SKOS information would advance in the right direction towards achieving the goals of this work.

3 Attaining semantic interoperability

In section 1, two disadvantages of the current state of SKOS were pointed out: lack of formalization of the terms in the SKOS Core, and the need for mapping criteria to foster the semantic interoperability between thesauri. The first problem will be addressed by proposing a set of precise definitions for the terms in the SKOS vocabulary through mapping them to terms in an upper ontology (*OpenCyc* is used here as a case study). This is a general mapping but also particular mappings stating something like "this term means this in this context" can also be carried out.

The second problem is addressed by defining an intermediate model to map the concepts in a SKOS scheme to terms already included in upper ontologies (again, *OpenCyc* is used here), as well as explicitly defining the terms in the SKOS vocabulary using an ontology language will improve the effective integration of semantically heterogeneous thesauri. This would foster the automated or semi-automated processing of SKOS schemes by Semantic Web applications in specific contexts of use, respecting the original SKOS information.

3.1 Providing formal definitions for SKOS metadata elements

Metadata elements in the SKOS vocabulary are divided into six categories: *conceptual elements*, *labelling properties*, *documentation properties*, *semantic relationships*, *collections* and *subject indexing elements*. In particular, the *Semantic Relationships* properties are metadata elements aimed at "asserting semantic (paradigmatic) relationships between concepts". However, SKOS does not provide a clear definition of what a semantic relationship is. In addition, some of the relationships are described in a purposefully vague language. Table 2 shows the relationship elements in SKOS: Although each relationship in the table above is informally defined in the SKOS Guide, the lack of

Table 2. SKOS relations.

SKOS element	**Definition**	**Comment**
`semanticRelation`	A relation of meaning	Not to be used directly, but as a super-property for all properties denoting a relationship of meaning
`narrower`	The scope (meaning) of one concept falls completely within the scope of another	Narrower concepts are typically rendered as children in a concept hierarchy tree
`broader`	A concept that is more general in meaning than another	Broader concepts are typically rendered as parents in a concept hierarchy (tree).
`related`	(weak semantics) A concept with which there is an associative semantic relationship	Expresses the fact that two concepts are in some way related, and that the relationship should not be used to create a hierarchy

Table 3. SKOS relations mapped to *OpenCyc* predicates.

SKOS relation	***OpenCyc* term**	**Comments**
`semanticRelation`	`Predicate`	—
`narrower`	`genls`	Relates a given collection to those collections that subsume it
	`subSet`	Relates a set or collection SUB to a set or collection SUPER whenever the extent (see extent) of SUB is a subset of the extent of SUPER
	`TaxonomicPredicate`	Used to help specify the position of a thing within one of the major taxonomies or hierarchies in the *OpenCyc* ontology
`broader`	inverse of `genls`	Relates the subsumed collection to the subsumer collection
`related`	not `TaxonomicPredicate`	—

formal definitions for every term hampers their use by Semantic Web applications. To perform reasoning tasks on the knowledge defined, avoiding misinterpretation of terms across different thesauri, computational agents require machine-readable descriptions (in addition to the human-readable versions of

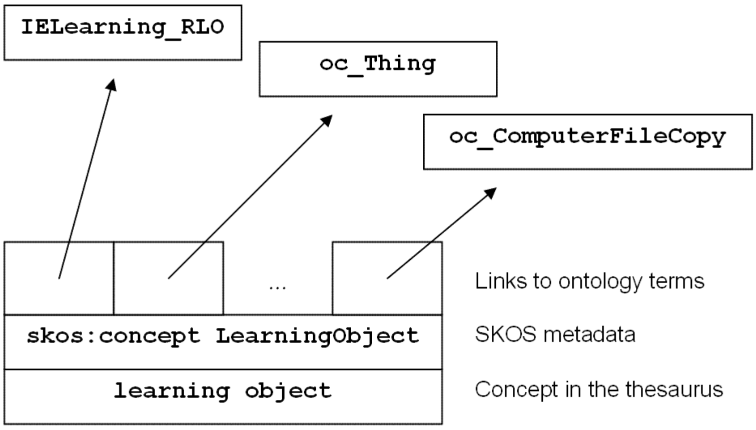

Figure 2. Mapping SKOS terms to upper ontologies.

the information). These can be provided in the form of explicit definitions in an ontology language such as OWL. Table 3 is a partial example on the effort to map these relationships to formally defined predicates in the *OpenCyc* knowledge base. Doing this, we are providing SKOS elements with a machine-consumption semantics that will disambiguate any interpretation.

3.2 An intermediate model to map SKOS terms to upper ontologies

The same concept can have different meanings across different thesauri. To both avoid misinterpretations and foster automated reasoning on the terms of the thesauri, terms can be linked to a general knowledge ontology, such as *OpenCyc*. This should be made without the need of modifying existing SKOS records for existing SKOS concept schemes. Figure 2 depicts how this can be done through a non-invasive intermediate model. In the example, a concept in a particular thesaurus, `learning object` [11], can be linked to specific meanings depending on the different characterizations of the concept (an interesting study on the existence of different learning object conceptualizations is that of McGreal [7]). The example assumes that an SKOS scheme has been created from the original concepts in the thesaurus. On top of this information, and probably performed by other persons (experts in upper ontologies or in learning ontologies), a number of intermediate records can be created to link this concept to terms formally defined in an ontology.

In this particular case, one organization could define learning object as "anything and everything" and thus link this concept to the term `Thing` in *OpenCyc* (the prefix "`oc`" indicates that it is an *OpenCyc* term), which is the top concept from which all the others derive. On the other hand, if we consider learning objects to be digital entities, they could be considered to be instances of `ComputerFileCopy` in *OpenCyc*, i.e. "information bearing things that contain digitally coded information readable by a computer". Although this definition is controversial due to the dynamic nature of many learning objects, it serves the purpose of abstracting them as elements available at a given URI. Finally, an organization mantaining a domain ontology on learning terms (such as the IE Research Unit, http://www.cc.uah.es/ie/) could be useful to link the concept learning object to, for example, the term `RLO` in its ontology, standing for reusable learning object. The prefix "`IELearning`" in the figure again indicates the origin of the ontology term.

To link terms in a SKOS scheme to *OpenCyc* terms a method described elsewhere [1] can be used. This process can be roughly described in these steps:

1. Find one or several terms that subsume the category under consideration.

2. Check carefully that the mapping is consistent with the rest of the subsumers inside *OpenCyc*.

3. Provide the appropriate predicates to characterize the new category.

4. Edit it in an ontology editor to come up with the final formal version.

This process has the advantage of allowing the individual work of an expert, whose outcomes can then be contrasted with the work of others. The results of the process are much more efficient and structured than engineering a new ontology, since the argumentation against or in favour of a given concept or predicate is put in the formal context of an upper ontology.

4 Conclusions and further research directions

The effort here described is the first step to the description of an intermediate model oriented to foster semantic interoperability among thesauri. Herein, only the SKOS semantic relationships have been described, as this is a part particularly useful for illustration purposes. However, the full description of the intermediate model should completely cover SKOS Core.

Regarding the use of upper ontology terms from *OpenCyc*, the opinion of other experts will be required in order to validate the links between the SKOS elements and the corresponding *OpenCyc* classes and properties. The authors consider these opinions very valuable and thus are open to positive feedback on the precision and usefulness of the definitions included.

Acknowledgments

This work is supported by the University of Alcalá (project UAH-PI2005-070) and the Spanish Ministry of Education (project TSI2004-21263-E).

References

1. Abran, A., Cuadrado, J.J., Garcia-Barriocanal, E., Mendes, O., Sanchez-Alonso, S. and Sicilia, M.A.: Engineering the ontology for the Software Engineering Body of Knowledge: Issues and Techniques, in Ontologies for software engineering, Springer Verlag (in press)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D. and Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.
3. Corazzon, R.: Ontology. A resource guide for philosophers. Retrieved November 1, 2005, from http://www.formalontology.it
4. Doan, A., Noy, N. and Haleevy, A.: Introduction to the special issue on semantic integration. SIGMOD Record, **33** (4) (2004).
5. Gruber, T. R.: A translation approach to portable ontologies. Knowledge Acquisition. **5**(2) (1993) 199–220.
6. Lenat, D. B.: Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM, **38**(11) (1995) 33–38.
7. McGreal, R.: Learning Objects: A Practical definition. International Journal of Instructional Technology and Distance Learning **1**(9) (2004).
8. Miles, A. and Brickley, D. (editors), SKOS Core Guide: W3C Working Draft 10. Retrieved November 1, 2005, from: http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/
9. Miles, A. and Brickley, D. (editors), SKOS Mapping Vocabulary Specification. Retrieved November 1, 2005, from: http://www.w3.org/2004/02/skos/mapping/spec/
10. Miles, A. and Brickley, D. (editors), SKOS Core Vocabulary Specification: W3C Working Draft 10. Retrieved November 1, 2005, from: http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20050510/
11. Polsani, P.R.: The Use and Abuse of Reusable Learning Objects. Journal of Digital information, **3** (4) (2002).