

SKOS: Simple Knowledge Organisation for the Web

Alistair Miles
CCLRC Rutherford Appleton Laboratory
a.j.miles@rl.ac.uk

José R. Pérez-Agüera
Department of Software Engineering and Artificial
Intelligence UCM.
jose.aguera@fdi.ucm.es

Abstract:

This article introduces the Simple Knowledge Organisation System (SKOS), a Semantic Web language for representing controlled structured vocabularies, including thesauri, classification schemes, subject heading systems and taxonomies. SKOS provides a framework for publishing a thesaurus or classification scheme, and for publishing a subject index over a collection of items (e.g. books), in the Semantic Web. By publishing in the Semantic Web, applications may harvest and merge these data, allowing them for example to implement a combined retrieval service across multiple collections (e.g. libraries).

This article also describes some alternatives for integrating Semantic Web services based on the Resource Description Framework (RDF) and SKOS into a distributed enterprise architecture.

Keywords:

Knowledge organization systems, KOS, taxonomies, thesauri, classification schemes, glossaries, RDF, OWL, Semantic Web, web services, software agents.

1. Introduction

SKOS (1, 2) is a formal language for representing a controlled structured vocabulary. By “controlled structured vocabulary” we mean to include:

- **Thesauri** broadly conforming to the ISO 2788:1986 guidelines such as the UK Archival Thesaurus (UKAT), the General Multilingual Environmental Thesaurus (GEMET), and the Art and Architecture Thesaurus (AAT).
- **Classification schemes** such as the Dewey Decimal Classification (DDC), the Universal Decimal Classification (UDC), and the Bliss Classification (BC2).
- **Subject heading systems** such as the Library of Congress Subject Headings (LCSH) and the Medical Subject Headings (MeSH).

A key feature of these vocabularies is that they are intended for use within information retrieval applications. I.e. they are used to describe items in a collection in a controlled way, allowing semantically precise and unambiguous retrieval.

SKOS is an application of the Resource Description Framework (RDF). Because RDF is a formal language that has well defined logical properties, any controlled structured vocabulary represented using SKOS is machine-understandable. I.e. a computer application can read it, “make sense” of it, and use it to provide functionalities such as rich visual search and browse user interfaces.

RDF is also the language of the Semantic Web. By design, RDF supports the distributed publication of data. This means that a controlled structured vocabulary published using SKOS can be linked to and/or merged with other data sources, such as subject indexes, or other vocabularies. This enables, for example, the typical situation where a retrieval service is required across a number of separately maintained collections, to be implemented without complex database integration.

SKOS is under active development at the time of writing, within the scope of the W3C's Semantic Web Best Practices and Deployment Working Group. The SKOS Core Guide (1) and the SKOS Core Vocabulary Specification (2), the two primary documents describing the usage of SKOS, are published as W3C Working Drafts.

This article gives three examples of the use of SKOS: a category taxonomy, a thesaurus, and a classification scheme. The examples are given in the RDF/XML serialization syntax, although note that RDF may be serialized according to other syntaxes such as Turtle/N3. In prose, prefixes such as "skos:" are used to abbreviate URIs. The following table gives the prefix conventions used.

Prefix	URI
skos:	http://www.w3.org/2004/02/skos/core#
dc:	http://purl.org/dc/elements/1.1/
dct:	http://purl.org/dc/terms/
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#

Abbreviated URIs are underlined, for example, skos:prefLabel stands for the URI "http://www.w3.org/2004/02/skos/core#prefLabel". Note also that the xml:base attribute provides a URI base for relative URIs within an RDF/XML document.

2. A Weblog Taxonomy in RDF

```
<rdf:RDF
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.wasab.dk/morten/blog/archives/author/mortenf/skos.rdf">

  <skos:ConceptScheme rdf:about="#scheme">
    <dc:title>Morten Frederiksen's Categories</dc:title>
    <dc:description>Concepts from the weblog "Binary Relations" based on category usage by
Morten Frederiksen.</dc:description>
    <dc:creator>Morten Frederiksen</dc:creator>
  </skos:ConceptScheme>

  <skos:Concept rdf:about="#c1">
    <skos:prefLabel xml:lang="en">General</skos:prefLabel>
    <skos:narrower rdf:resource="#c23"/>
    <skos:narrower rdf:resource="#c30"/>
    <skos:inScheme rdf:resource="#scheme"/>
  </skos:Concept>

  <skos:Concept rdf:about="#c23">
    <skos:prefLabel xml:lang="en">Travelling</skos:prefLabel>
    <skos:broader rdf:resource="#c1"/>
    <skos:inScheme rdf:resource="#scheme"/>
  </skos:Concept>

  <skos:Concept rdf:about="#c30">
    <skos:prefLabel xml:lang="en">Politics</skos:prefLabel>
    <skos:broader rdf:resource="#c1"/>
    <skos:inScheme rdf:resource="#scheme"/>
  </skos:Concept>

</rdf:RDF>
```

The example above is adapted from Morten Frederiksen's web log categories. Morten's categories are arranged in a category hierarchy, and include the following:

General Travelling Politics

Morten uses these categories to organise the entries in his web log – each entry can belong to one or more categories.

The SKOS representation of these categories is shown above. Each category is represented as a resource of type `skos:Concept` – this class is the basic building block of all SKOS descriptions. Each concept has a preferred label, given by the `skos:prefLabel` property. Note that the language (“en”) has been given for the preferred labels – a concept may be given labels and annotations in multiple languages. Concepts are linked to other concepts via “semantic relation” properties, in this case the `skos:broader` and `skos:narrower` properties indicating a generalization/specialization relationship. Finally, the concepts are all part of a “concept scheme”, represented as a resource of type `skos:ConceptScheme`. The concepts are linked to the concept scheme in which they participate by the `skos:inScheme` property. Some descriptive metadata is also given using the Dublin Core Metadata Terms.

3. A Thesaurus in RDF

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.ukat.org.uk/thesaurus/concept/">

  <skos:ConceptScheme rdf:about="http://www.ukat.org.uk/thesaurus">
    <dc:title>The UK Archival Thesaurus</dc:title>
  </skos:ConceptScheme>

  <skos:Concept rdf:about="1750">
    <skos:prefLabel xml:lang="en">Economic cooperation</skos:prefLabel>
    <skos:altLabel xml:lang="en">Economic co-operation</skos:altLabel>
    <skos:scopeNote>Includes cooperative measures in banking, trade, industry etc., between
and among countries.</skos:scopeNote>
    <skos:broader rdf:resource="4382"/>
    <skos:narrower rdf:resource="2108"/>
    <skos:narrower rdf:resource="9505"/>
    <skos:narrower rdf:resource="15053"/>
    <skos:narrower rdf:resource="18987"/>
    <skos:related rdf:resource="3250"/>
    <skos:inScheme rdf:resource="http://www.ukat.org.uk/thesaurus"/>
  </skos:Concept>

</rdf:RDF>
```

The example above is adapted from an extract of the UK Archival Thesaurus (18). The extract would typically be presented in the traditional thesaurus style is as follows:

Economic cooperation

- UF Economic co-operation
- BT Economic policy
- NT Economic integration
- NT European economic cooperation
- NT European industrial cooperation
- NT Industrial cooperation
- RT Interdependence
- SN Includes cooperative measures in banking, trade, industry etc., between and among countries.

The UKAT broadly conforms to the ISO 2788:1986 guidelines for the development and construction of thesauri. When representing such a thesaurus using SKOS, a URI must first be allocated to each of the *conceptual units* of the thesaurus, that is, to each unit of the thesaurus that has a distinct meaning. In this example, the URI “<http://www.ukat.org.uk/thesaurus/concept/1750>” has been allocated to the conceptual unit established by the descriptor “Economic cooperation”. The use of URIs to uniquely and unambiguously refer to precise meanings is of course essential in a Semantic Web context, where data is being aggregated and merged from multiple sources.

The lexical value of a descriptor (i.e. the character string) is mapped to the literal value of the `skos:prefLabel` property in the appropriate language (in this case “en”). The lexical values of any non-descriptors are mapped to literal values of the `skos:altLabel` (“alternative label”) property. Any annotations such as scope notes are mapped to the appropriate documentation property, in this case the `skos:scopeNote` property is used. Other documentation properties available include `skos:definition`, `skos:historyNote` and `skos:editorialNote`. It is also possible to define custom documentation properties via the extensibility mechanisms used in SKOS (see below).

As in the previous example, the `skos:narrower` and `skos:broader` semantic relation properties have been used to indicate generalization/specialization relationships. Also, the `skos:related` semantic relation property has been used – this property indicates an associative relationship between two concepts. Again, it is possible to define custom semantic relation properties via extensibility.

The example above only shows the RDF description of the single “Economic cooperation” concept. Of course, each of the narrower, broader and related concepts linked from this concept would have its own RDF description, and these would all form part of the RDF description of the thesaurus as a whole.

A thesaurus such as the UKAT is typically used to construct a subject index over a collection of items. That is, the thesaurus is used to describe the subject matter of the items in a consistent, precise and unambiguous way, which then enables precise and unambiguous searching. SKOS includes the properties `skos:subject` and `skos:primarySubject` for describing the subject matter of resources in a controlled way. The `skos:subject` property is very similar to the `dc:subject` property (in fact it is a sub-property of the latter), however whereas the range of the `dc:subject` property is unconstrained, only resources of type `skos:Concept` may act as values to the `skos:subject` property. For example, the RDF/XML snippet below uses the “Economic cooperation” concept from the UKAT to describe the subject matter of a web page.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#">

  <rdf:Description rdf:about="http://www.example.com/money.html">
    <skos:subject rdf:resource="http://www.ukat.org.uk/thesaurus/concept/1750"/>
  </rdf:Description>

</rdf:RDF>
```

The `skos:primarySubject` property is a sub-property of `skos:subject`, and allows you to distinguish one concept as the primary or principal subject, where the subject matter of an item covers several concepts.

4. A Classification Scheme in RDF

The example below is adapted from an extract of the Physics and Astronomy Classification Scheme (PACS).

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs/" >

  <skos:ConceptScheme rdf:about="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs">
    <dc:title>Physics and Astronomy Classification Scheme</dc:title>
    <dc:creator>American Institute of Physics</dc:creator>
    <skos:hasTopConcept rdf:resource="90."/>
  </skos:ConceptScheme>

  <skos:Concept rdf:about="90.">
    <skos:prefLabel>GEOPHYSICS, ASTRONOMY, AND ASTROPHYSICS</skos:prefLabel>
    <skos:narrower rdf:resource="91."/>
    <skos:inScheme rdf:resource="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs"/>
  </skos:Concept>

  <skos:Concept rdf:about="91.">
    <skos:prefLabel>Solid Earth physics</skos:prefLabel>
    <skos:broader rdf:resource="90."/>
    <skos:narrower rdf:resource="91.10.-v"/>
    <skos:inScheme rdf:resource="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs"/>
  </skos:Concept>

  <skos:Concept rdf:about="91.10.-v">
    <skos:prefLabel>Geodesy and gravity</skos:prefLabel>
    <skos:broader rdf:resource="91."/>
    <skos:narrower rdf:resource="91.10.Pp"/>
    <skos:inScheme rdf:resource="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs"/>
  </skos:Concept>

  <skos:Concept rdf:about="91.10.Pp">
    <skos:prefLabel>Gravimetric measurements and instruments</skos:prefLabel>
    <skos:broader rdf:resource="91.10.-v"/>
    <skos:inScheme rdf:resource="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs"/>
  </skos:Concept>

</rdf:RDF>

```

This extract would normally be presented as follows:

90. GEOPHYSICS, ASTRONOMY, AND ASTROPHYSICS

91. Solid Earth physics

91.10.-v Geodesy and gravity

91.10.Pp Gravimetric measurements and instruments

Because each “class” or “category” in a classification scheme establishes a distinct meaning, a URI must be allocated for each. For example, in the above the URI “<http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs/91.10.-v>” has been allocated to the “Geodesy and gravity” category.

Although the conceptual units of a classification scheme are sometimes referred to as “classes” they are usually not “classes” in the stricter logical sense, although they are used to “classify” documents by their subject matter. Also, the hierarchical relationships between them are usually not strict class subsumption relationships. Hence it is inappropriate to model a classification scheme as a subsumption hierarchy of RDFS or OWL classes. The SKOS notions of a “concept” and of a “broader/narrower” generalization hierarchy are a better fit, and help to avoid unexpected or inappropriate inferences being drawn by RDFS or OWL reasoners.

The example above uses the same features previously described in other examples. Namely the skos:Concept and skos:ConceptScheme classes as the basic building blocks, the skos:prefLabel labelling property, and the skos:broader and skos:narrower semantic relation properties. This example also introduces the skos:hasTopConcept property, which is used to explicitly indicate

which concepts are the topmost in the generalization hierarchy.

As with thesauri, the typical use of a classification scheme is to describe the subject matter of some items in a controlled and unambiguous way. However, whereas a thesaurus is typically used to describe one or more subjects of an item, a classification scheme is used to classify items by their *primary subject*. Hence, when representing a subject classification in RDF, use the `skos:primarySubject` property. Additional subjects can then be added using the `skos:subject` property, without losing the unique classification by primary subject.

The snippet below illustrates the use of the `skos:primarySubject` property to represent a subject classification of an item.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#">

  <rdf:Description rdf:about="http://www.example.com/all-about-gravity.html">
    <skos:primarySubject
      rdf:resource="http://www.w3.org/2001/sw/Europe/reports/thes/ns/pacs/91.10.-v"/>
  </rdf:Description>

</rdf:RDF>
```

5. Extending SKOS

SKOS can be extended by refinement. This means that third parties can declare and use classes and/or properties that *refine* (i.e. are sub-classes or sub-properties of) SKOS classes or properties. The valid RDFS inferences can then be applied by anyone to *infer* a pure SKOS representation of a controlled structured vocabulary from a representation that uses custom refinement extensions. This strategy guarantees backwards compatibility for refinement extensions.

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix x: <http://www.example.com/skos-extension#>.

x:prefScientificLabel a rdf:Property;
  rdfs:label 'preferred scientific label';
  rdfs:comment 'The preferred lexical label for scientists.'@en;
  rdfs:subPropertyOf skos:prefLabel;
.

x:altScientificLabel a rdf:Property;
  rdfs:label 'alternative scientific label';
  rdfs:comment 'An alternative lexical label for scientists.'@en;
  rdfs:subPropertyOf skos:altLabel;
.

x:prefNonScientificLabel a rdf:Property;
  rdfs:label 'preferred non-scientific label';
  rdfs:comment 'The preferred lexical label for non-scientists.'@en;
  rdfs:subPropertyOf skos:altLabel;
.

x:altNonScientificLabel a rdf:Property;
  rdfs:label 'alternative non-scientific label';
  rdfs:comment 'An alternative lexical label for non-scientists.'@en;
  rdfs:subPropertyOf skos:altLabel;
.
```

Any of the classes or properties in the SKOS Core Vocabulary can be extended in this way. For example, a controlled vocabulary might require different labels for different audiences, e.g. scientific and non-scientific users. In this case we can declare refinement extensions to the SKOS labelling properties `skos:prefLabel` and `skos:altLabel`, as shown above (N.B. the examples in this

section use the Turtle syntax for RDF).

Once declared, the extensions can be used, as shown in the following box. In the example given, the custom labelling properties are used to attach scientific and non-scientific labels to the concept of “aspirin”.

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix x: <http://www.example.com/skos-extension#>.
@prefix eg: <http://www.example.com/thesaurus#>.

eg:concept002 a skos:Concept;
  x:prefScientificLabel 'acetylsalicylic acid'@en;
  x:altScientificLabel '2-acetoxybenzoic acid'@en;
  x:prefNonScientificLabel 'aspirin'@en;
```

By drawing valid RDFS conclusions, we can then derive a pure SKOS representation of the same concept, as shown below. Note that only one of the four extension properties was declared as a sub-property of `skos:prefLabel` – this ensured that, after all valid inferences are made, there is still only one preferred lexical label in any given language (it wouldn't make sense for something to have more than one “preferred” label). When declaring refinement extensions, some care is required to ensure that the conclusions they entail are all sensible.

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix eg: <http://www.example.com/thesaurus#>.

eg:concept002 a skos:Concept;
  skos:prefLabel 'acetylsalicylic acid'@en;
  skos:altLabel '2-acetoxybenzoic acid'@en;
  skos:altLabel 'aspirin'@en;
```

The same basic extension mechanism can also be used to construct hybrid SKOS/OWL ontologies – that is, structured vocabularies that have some features of an ontology and some features of a thesaurus or classification scheme. Such hybrid ontologies are a compromise between the cost of creating and maintaining more precise semantics and the benefits of richer search and browse applications. The example on the following page is adapted from the Semantic Web Environmental Directory (SWED).

6. Integration Within an Enterprise Architecture

How can SKOS support the use of controlled structured vocabularies within a distributed enterprise software architecture?

The simplest way to make a controlled vocabulary available within a distributed environment is to publish the SKOS representation of the entire vocabulary as a single RDF/XML document on an HTTP server. Any other component within the enterprise can then simply retrieve the vocabulary by issuing an HTTP GET request. However, if the vocabulary is large, this may not be a practical solution, because it forces every component to retrieve the entire vocabulary, even if it only needs a small part of it.

To solve this problem, the SKOS representation of the vocabulary can be made available via a SPARQL service. SPARQL Query is an RDF query language (W3C Candidate Recommendation at the time of writing), and allows data from one or more RDF graphs to be queried and selected, in a similar way to the relational query language SQL. The SPARQL Protocol gives network protocol bindings for a SPARQL service, i.e. it describes the network interactions that can be used to query a data source and obtain results. RDF toolkits can be expected to support SPARQL Query and Protocol off the shelf, which greatly simplifies the task of deploying a query service for a controlled vocabulary.

```

@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix x: <http://www.example.com/ontology#>.

## Set up the framework

x:EnvironmentalOrganisation a owl:Class;
  rdfs:label 'Environmental Organisation';
  rdfs:comment 'The class of organisations whose core business has something to do with
the natural environment.'.

x:topicOfInterest a owl:ObjectProperty;
  rdfs:label 'topic of interest';
  rdfs:comment 'The main topic of interest of an organisation.';
  rdfs:domain x:EnvironmentalOrganisation;
  rdfs:range x:TopicOfInterestConcept.

x:TopicOfInterestConcept a owl:Class;
  rdfs:label 'Topic of Interest Concept';
  rdfs:comment 'The class of concepts used to describe the topic of interest of an
organisation.';
  rdfs:subClassOf skos:Concept.

x:activity a owl:ObjectProperty;
  rdfs:label 'activity';
  rdfs:comment 'The primary activity of an organisation.';
  rdfs:domain x:EnvironmentalOrganisation;
  rdfs:range x:ActivityConcept.

x:ActivityConcept a owl:Class;
  rdfs:label 'Activity Concept';
  rdfs:comment 'The class of concepts used to describe the activity of an organisation.';
  rdfs:subClassOf skos:Concept.

# Declare some activity and topic-of-interest concepts

x:topicConcept001 a x:TopicOfInterestConcept;
  skos:prefLabel 'Sustainable development';
  skos:narrower x:topicConcept010.

x:topicConcept010 a x:TopicOfInterestConcept;
  skos:prefLabel 'Ecotourism';
  skos:broader x:topicConcept001.

x:activityConcept002 a x:ActivityConcept;
  skos:prefLabel 'Education and training';
  skos:narrower x:activityConcept012.

x:activityConcept012 a x:ActivityConcept;
  skos:prefLabel 'Awareness raising';
  skos:broader x:activityConcept002.

# Now use all the above to describe an environmental organisation

x:org1 a x:EnvironmentalOrganisation;
  rdfs:label 'Society for Environmental Exploration (SEE)';
  x:topicOfInterest x:topicConcept001;
  x:activity x:activityConcept012;
  foaf:homepage <http://www.frontier.ac.uk/>.

```

A potential drawback to using a SPARQL service may be that, because SPARQL is a very general RDF query language, the query processor may not be optimized for tasks specific to a particular type of data and application. Controlled structured vocabularies are typically used within information retrieval applications, and therefore a similar set of operations is likely to be repeated. Also, for information retrieval applications, response time is an important factor in system acceptance, and therefore some form of optimization may be necessary.

An optimized data service for controlled vocabularies might use a number of techniques to provide fast and efficient access to the data. For example, hash tables could be used to map string values of lexical labels to programmatic objects representing concepts, hence providing a very efficient

“entry” into a vocabulary structure. Because the functionalities are specialized, they are typically encapsulated within a custom programmatic interface, which could then be bound to concrete network and interaction protocols such as HTTP and SOAP. Notably, it is quite reasonable to use an RDF encoding with e.g. SOAP messages, and therefore SKOS and RDF can be used as a basis for data encoding at the interface level. This type of solution provides a compromise between the need to provide specialized functionality and the use of standard data encodings and interaction protocols.

7. Summary and Conclusions

This article has introduced the Simple Knowledge Organisation System (SKOS), with examples of its use for the RDF representation of three types of controlled structured vocabulary: a taxonomy, a thesaurus, and a classification scheme. Controlled vocabularies are typically used to describe the subject matter of information resources in a consistent and unambiguous way, to enable semantically precise retrieval, and SKOS also provides basic support for representing a subject index or subject classification over a collection of items.

SKOS is, at the time of writing, a work in progress, and a number of outstanding issues remain. Although SKOS may become a W3C Recommendation, development is likely to continue until December 2007 before this may be achieved. Therefore, all comments, suggestions and feedback relating to practical experience and/or theoretical considerations are warmly welcomed, and should be sent to the public-esw-thes@w3.org mailing list.

8. Acknowledgements

The contributions of the members of the Semantic Web Interest Group mailing list public-esw-thes@w3.org are gratefully acknowledged.

9. References

- A. J. Miles and D. Brickley eds. SKOS Core Guide, W3C Editor's Working Draft 15 February 2005. World Wide Web Consortium, 2005.
- A. J. Miles and D. Brickley eds. SKOS Core Vocabulary Specification, W3C Editor's Working Draft 2005-04-27. World Wide Web Consortium, 2005.
- G. Klyne and J. J. Carroll eds. Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- P. Hayes ed. RDF Semantics, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- D. Beckett ed. RDF/XML Syntax Specification (Revised), W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- D. Brickley and R. V. Guha eds. RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- F. Manola and E. Miller eds. RDF Primer, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- Semantic Web Best Practices and Deployment Working Group homepage, see <http://www.w3.org/2001/sw/BestPractices/>.
- D. L. McGuinness and F. van Harmelen eds. OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- M. K. Smith, C. Welty, D. L. McGuinness eds. OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- M. Dean and G. Schreiber eds. OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004. World Wide Web Consortium, 2004.
- DCMI Usage Board. DCMI Metadata Terms 2005-01-10. DCMI Recommendation, Dublin Core Metadata Initiative, 2005.

S. Kokkelink and R. Schwanzl. Expressing Qualified Dublin Core in RDF / XML 2002-05-15. DCMI Proposed Recommendation, Dublin Core Metadata Initiative, 2002.

A. Powell and H. Wagner eds. Namespace Policy for the Dublin Core Metadata Initiative (DCMI) 2001-10-26. DCMI Recommendation, Dublin Core Metadata Initiative, 2001.

DCMI Usage Board. DCMI Grammatical Principles 2003-11-18. Dublin Core Metadata Initiative, 2003.

Glossary from World Wide Web Consortium Process Document in RDF. See <<http://www.w3.org/2003/03/glossary-project/data/glossaries/Process.rdf>>. W3C Glossary and Dictionary Project. World Wide Web Consortium 2004.

M. Frederiksen. SKOS Output from Binary Relations Weblog. See <<http://www.wasab.dk/morten/blog/archives/author/mortenf/skos.rdf>>.

The UK Archival Thesaurus 2004. See <<http://www.ukat.org.uk/>>.

BS8723, Structured Vocabularies for Information Retrieval. BSI Public Draft, 2004.

Documentation - Guidelines for the Establishment and Development of Monolingual Thesauri, 2nd ed., ISO 2788:1986.

Documentation - Guidelines for the Establishment and Development of Multilingual Thesauri, 2nd ed., ISO 5964:1985.

The Semantic Web Advanced Development for Europe project homepage. See <<http://www.w3.org/2001/sw/Europe/>>.

A. J. Miles. RDF Encoding of Classification Schemes, SWAD-Europe Deliverable 8.5. World Wide Web Consortium 2004.

B. M. Matthews and A. J. Miles. Review of RDF Thesaurus Work, A review and discussion of RDF schemas for thesauri, SWAD-Europe Deliverable 8.2. World Wide Web Consortium 2003.

American Institute of Physics. Physics and Astronomy Classification Scheme 2003. American Institute of Physics 2003.

D. Brickley, L. Miller. FOAF Vocabulary Specification, Namespace Document 3 April 2005 - (Back In Business Edition).

P. Cross, D. Brickley, T. Koch. RDF Thesaurus Specification (draft). ILRT Technical Report Number 1011. Intitute for Learning Research Technology, 2001.

B. M. Matthews, M. D. Wilson, K. Miller, J. Ryssevik. Internationalising Data Access Through LIMBER. Proc. third international workshop on internationalisation of products and systems, 2001.

I. Jacobs ed. World Wide Web Consortium Process Document 5 February 2004. World Wide Web Consortium, 2004.

Mail Archives for public-esw-thes@w3.org see <<http://lists.w3.org/Archives/Public/public-esw-thes/>>.

10. Notes

[1] We can find an excellent review of the various thesaurus mark-up proposals in <http://www.w3c.rl.ac.uk/SWAD/deliverables/8.2.html#4.1>

[2] Available in <http://ceres.ca.gov/thesaurus/RDF.html>

[3] <http://ceres.ca.gov/thesaurus/>

[4] http://www.w3schools.com/rdf/rdf_owl.asp

[5] <http://www.niso.org/standards/standarddetail.cfm?stdid=518>

[6] <http://www.w3.org/TR/2004/REC-owl-features-20040210/>

[7] <http://www.mindswap.org/2003/CancerOntology/>

[8] Eva M. Méndez Rodríguez (2002) Metadatos y recuperación de información: estándares, problemas y aplicabilidad en bibliotecas digitales, Gijón, Trea

[9] <http://zthes.z3950.org/>

[10] <http://www.topicmaps.org/>

[11] The word class comes from the terminology used in object oriented programming. The application was implemented using Java, what explains the extensive use of object oriented terminology throughout this text.

[12] A String is a data type referring to any set of characters. In our implementation we use one array of characters for each term, because is faster than String Object in Java

[13] RDF generation was done using Jena 2.1

[14] Gil Urdiciain, Blanca (1996) Manual de lenguajes documentales, Madrid, Noesis, p. 215–220.

[15] <http://www.google.com/apis/>.

[16] Ogbuji, Uche, Using RDF with SOAP: beyond remote procedure calls,
<http://www-106.ibm.com/developerworks/webservices/library/ws-soaprdf/>