

SWAD-Europe Thesaurus Activity

Deliverable 8.1



An RDF Schema for Thesauri (SKOS-Core 1.0 Guide)

Description of an RDF Schema for thesauri and related knowledge organisation systems

N.B. This document series is now deprecated. Please see <http://www.w3.org/2004/02/skos/core/guide/> for the latest version of the [SKOS Core Guide](#).

Abstract:

This document is a guide to using the SKOS-Core 1.0 RDF schema for thesauri. The SKOS meta-model is introduced, the features of the SKOS-Core 1.0 schema are described with examples, and guidelines for using SKOS concept schemes for subject-based indexing of web resources are offered.

Project name:

Semantic Web Advanced Development for Europe (SWAD-Europe)

Project Number:

IST-2001-34732

Workpackage name:

8. Thesaurus Research Prototype

Workpackage description:

<http://www.w3.org/2001/sw/Europe/plan/workpackages/live/esw-wp-8.html>

Deliverable title:

8.1: thesaurus_schema_draft

This version:

<http://www.w3.org/2001/sw/Europe/reports/thes/8.1/version05.html>

Latest version:

<http://www.w3.org/2001/sw/Europe/reports/thes/8.1/>

Previous version:

<http://www.w3.org/2001/sw/Europe/reports/thes/8.1/version04.html>

Status:

Completed

Authors:

[Alistair J. Miles](#), CCLRC

[Nikki Rogers](#), ILRT

[Dave Beckett](#), ILRT

Status of this document

This section describes the status of this document at the time of its publication. This is a draft document and may be updated, replaced, or obsoleted by other documents at any time. The latest status of this document series is maintained at the W3C.

This document is a public DRAFT for discussion. This guide and the SKOS-Core schema are an output of the research work of the [Semantic Web Advanced Development for Europe Project](#), which is associated with the [W3C Semantic Web Activity](#). This document is made available by W3C for discussion only. Publication of this document by W3C does not imply endorsement by W3C, including the Team and Membership.

Comments on this document are welcome and should be sent to the authors or to the public-esw-thes@w3.org list. An archive of this list is available at <http://lists.w3.org/Archives/Public/public-esw-thes/>.

Contents

1. [Introduction](#)
 2. [The SKOS Meta-Model](#)
 3. [Using SKOS-Core 1.0](#)
 - 3.1. [Defining a concept scheme](#)
 - 3.2. [Defining concepts](#)
 - 3.3. [Labelling concepts](#)
 - 3.4. [External \(non-URI\) identifiers for concepts](#)
 - 3.5. [Adding scope-notes, definitions, contextual examples and depictions](#)
 - 3.6. [Using symbols to label concepts](#)
 - 3.7. [Basic semantic relations between concepts](#)
 - 3.7.1. [Broader / Narrower](#)
 - 3.7.2. [Related](#)
 - 3.8. [Multilingual labelling](#)
 - 3.9. [More precise semantics](#)
 - 3.9.1. [BroaderGeneric / NarrowerGeneric](#)
 - 3.9.2. [BroaderInstantive / NarrowerInstantive](#)
 - 3.9.3. [BroaderPartitive / NarrowerPartitive](#)
 - 3.9.4. [RelatedPartOf / RelatedHasPart](#)
 - 3.10. [High-level grouping of concepts](#)
 4. [Using SKOS-Core with Dublin Core for subject-based indexing](#)
 5. [Further Reading](#)
-

1. Introduction

[\[back to contents\]](#)

This document is a guide to accompany the [SKOS-Core 1.0](#) schema, which can be found under the namespace:

<http://www.w3.org/2004/02/skos/core#>

SKOS-Core 1.0 is an RDF schema for representing thesauri and similar types of knowledge organisation system (KOS). SKOS-Core can be used for:

- Porting existing KOS to the Semantic Web
- Building simple concept schemes for the Semantic Web from scratch

SKOS stands for Simple Knowledge Organisation Systems. SKOS-Core is intended as a complement to OWL. It does provide a basic framework for building concept schemes, but it does not carry the strictly defined semantics of OWL. Thus it is ideal for representing those types of KOS, such as thesauri, that cannot be mapped directly to an OWL ontology. SKOS is also easier to use, and harder to misuse than OWL, providing an ideal entry point for those wishing to use the Semantic Web for knowledge organisation.

SKOS-Core also provides a framework for linking concepts to the words and phrases that are normally used by people to refer to them. This valuable information, once captured, can be used to support a number of tasks, such as automated classification of web documents, and automated multilingual translation of glossaries.

This guide first describes the SKOS meta-model, and introduces the terminology that will be used throughout. There follows a description of the features of the SKOS-Core 1.0 schema, with examples of how to use them. Finally there are guidelines for using SKOS concept schemes for subject-based indexing of web resources.

2. The SKOS Meta-Model [\[←back to contents\]](#)

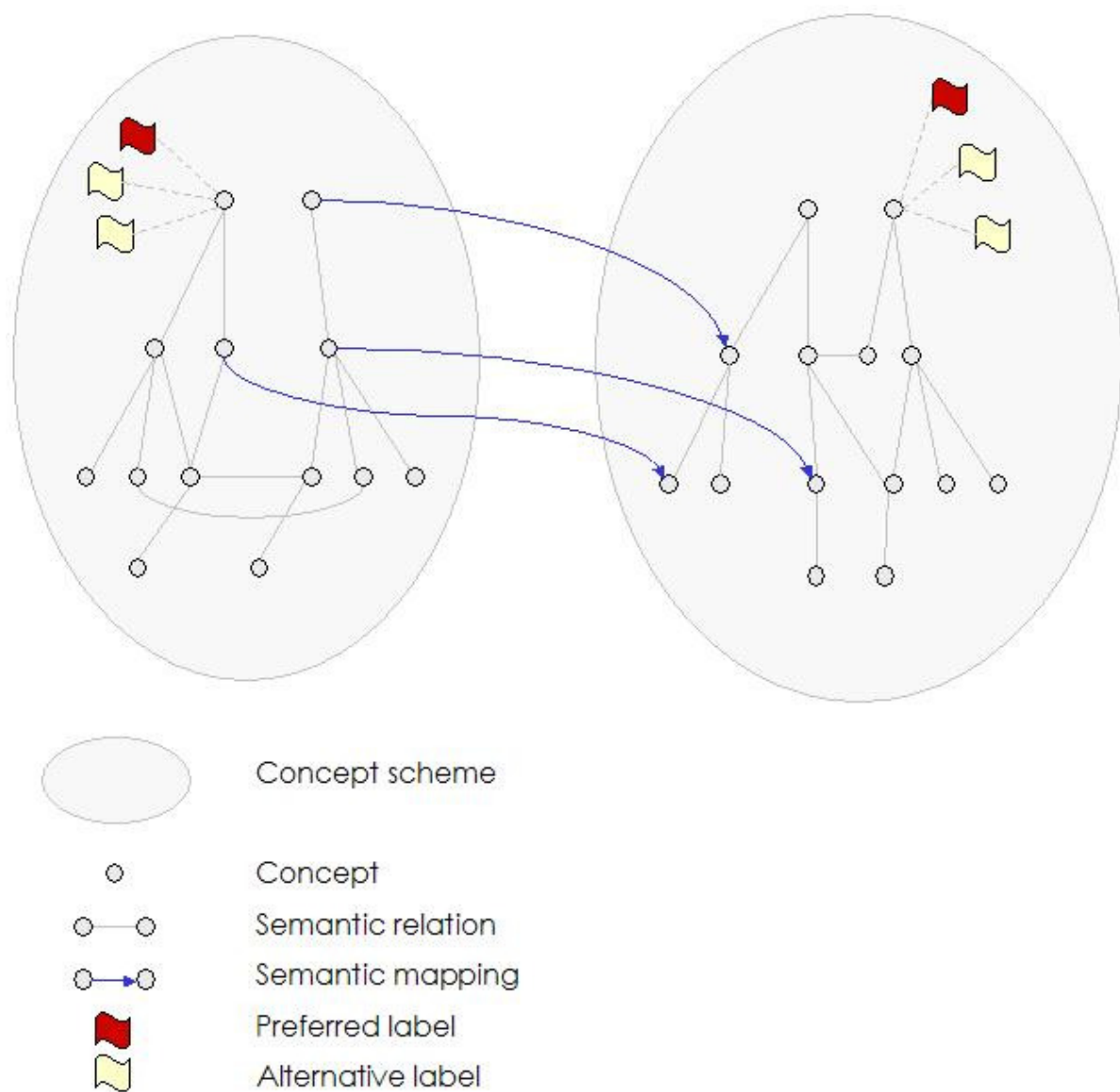
SKOS-Core allows you to define concepts and concept schemes. A **concept** is any unit of thought that can be defined or described. A **concept scheme** is a collection of concepts.

A concept may have any number of attached labels. A **label** is any word, phrase or symbol that can be used to refer to the concept by people. A concept may have only one preferred label, and any number of alternative labels.

Relationships may be defined between concepts within the same concept scheme. Any such relationship is referred to here as a **semantic relation**.

Mappings may be defined between concepts from different concept schemes. Any such mapping is referred to here as a **semantic mapping**.

Figure 1. The SKOS Meta-Model



3. Using SKOS-Core 1.0 [\[back to contents\]](#)

The examples below demonstrate how to use the SKOS-Core 1.0 schema to create a simple concept scheme in the style of a thesaurus.

3.1. Defining a concept scheme [\[back to contents\]](#) -

The first step is to describe the scheme itself. To allow unambiguous reference to your scheme in the future, give it a URI. Then state that it has type `skos:ConceptScheme`. To describe the meta-properties of the scheme, you can use e.g. Dublin Core, for example:

Example of a concept scheme declaration

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <skos:ConceptScheme rdf:about="http://example.com/thesaurus">
    <dc:title>The SWAD-Europe Example Thesaurus</dc:title>
    <dc:description>An example thesaurus to illustrate the use of the
    <dc:creator>A J Miles</dc:creator>
  </skos:ConceptScheme>

</rdf:RDF>
```

3.2. Defining concepts [\[←back to contents\]](#) -

For each concept you define, there must be some way for both people and machines to refer to that concept unambiguously in the future. The simplest way to ensure unambiguous global reference is to assign each concept a URI. Then state that the concept has type `skos:Concept`. For example:

Example of a concept declaration

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

  <skos:Concept rdf:about="http://example.com/Concept/0001"/>

</rdf:RDF>
```

A concept does not have to be a part of any scheme. It may also be a part of more than one scheme. To express that a concept is a part of a scheme, use the `skos:inScheme` property. For example:

Example of using `skos:inScheme`

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

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:Concept>

</rdf:RDF>
```

3.3. Labelling concepts [\[←back to contents\]](#) -

A concept may be given only one preferred label, and any number of alternative labels. For example:

Example of concept labelling

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

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:prefLabel>Bangers and mash</skos:prefLabel>
    <skos:altLabel>Sausage and mash</skos:altLabel>
    <skos:altLabel>Sausage and mashed potato</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:Concept>

</rdf:RDF>
```

To help other people unambiguously refer to your concepts without having to use URIs, it is strongly recommended that no two concepts in your scheme be given the same preferred label.

The alternative labels can be any set of words or phrases, including alternative spelling variants (and mis-spellings), that could be used to refer to the concept. In the traditional thesaurus sense, this set of labels constitutes the 'entry vocabulary' for your concept. It is there primarily to help other people navigate to the right concept. The better the coverage of possible labels for each concept, the more likely another person is to find the concept they want.

3.4. External (non-URI) identifiers for concepts [\[←back to contents\]](#) -

If a concept has already been defined outside the scope of the semantic web, it may have been given a unique identifier that is not a URI. To express this property of a concept, use

the `skos:externalID` property. For example:

Example of using `skos:externalID`

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

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:externalID>A.01.0001</skos:externalID>
    <skos:prefLabel>Bangers and mash</skos:prefLabel>
    <skos:altLabel>Sausage and mash</skos:altLabel>
    <skos:altLabel>Sausage and mashed potato</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:Concept>

</rdf:RDF>
```

3.5. Adding scope-notes, definitions, contextual examples and depictions [\[back to contents\]](#) -

SKOS-Core 1.0 has three properties to help you establish exactly what each concept is intended to mean. Use `skos:scopeNote` to attach any piece of text that helps to further elucidate the intended meaning of a concept. Use `skos:definition` to attach a formal (dictionary-style) definition to a concept. Use `skos:example` to attach a contextual example of the use of one of the labels (usually the preferred label). For example:

Example of using `skos:scopeNote`

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

  <skos:Concept rdf:about="http://example.com/Concept/0011">
    <skos:prefLabel>Marble</skos:prefLabel>
    <skos:scopeNote>A granular crystalline limestone.</skos:scopeNote>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:Concept></rdf:RDF>
```

Example of using `skos:definition`

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

  <skos:Concept rdf:about="http://example.com/Concept/0012">
    <skos:prefLabel>Liquid crystal</skos:prefLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:definition>A substance which flows like a liquid but has sc
    degree of ordering in the arrangement of its molecules.</skos:def
  </skos:Concept>

</rdf:RDF>
```

Example of using `skos:example`

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

  <skos:Concept rdf:about="http://example.com/Concept/00013">
    <skos:prefLabel>Medicate</skos:prefLabel>
    <skos:altLabel>Administer medicine</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:example>both infants were heavily medicated to alleviate th
  </skos:Concept>

</rdf:RDF>
```

It may also be helpful to attach to a concept images depicting it. This may be done using the

foaf:depiction property. For example:

Example of using foaf:depiction

```
<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:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Image rdf:about="http://example.com/img/bangersandmash.jpg"/>

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:prefLabel>Bangers and mash</skos:prefLabel>
    <skos:altLabel>Sausage and mash</skos:altLabel>
    <skos:altLabel>Sausage and mashed potato</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <foaf:depiction rdf:resource="http://example.com/img/bangersandmash.jpg"/>
  </skos:Concept>
```

3.6. Using symbols to label concepts [\[back to contents\]](#) -

If it is more appropriate for you to label your concepts with symbols, rather than words or phrases, use the `skos:prefSymbol` and `skos:altSymbol` properties. For example:

Example of using symbols to label concepts

```
<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:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Image rdf:about="http://example.com/symbols/0001.jpg"/>
  <foaf:Image rdf:about="http://example.com/symbols/0001a.jpg"/>
  <foaf:Image rdf:about="http://example.com/symbols/0001b.jpg"/>

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:prefSymbol rdf:resource="http://example.com/symbols/0001.jpg"/>
    <skos:altSymbol rdf:resource="http://example.com/symbols/0001a.jpg"/>
    <skos:altSymbol rdf:resource="http://example.com/symbols/0001b.jpg"/>
  </skos:Concept>

</rdf:RDF>
```

3.7. Basic semantic relations between concepts [\[back to contents\]](#) -

SKOS-Core contains a family of properties for expressing simple relationships between concepts within a concept scheme. The top-level property of this family is the `skos:semanticRelation` property. Below this are the `skos:broader`, `skos:narrower` and `skos:related` properties.

3.7.1. Broader / Narrower

skos:broader/skos:narrower property hierarchy

skos:semanticRelation	skos:semanticRelation
+--skos:broader	+--skos:narrower

The `skos:broader` and `skos:narrower` properties allow you to create a hierarchy of concepts. The properties themselves carry weak semantics - they should be used to express the fact that a concept is in some way more or less general than another. I.e. they imply that the concepts involved may reasonably be arranged into a hierarchy, without being too strict about the exact meaning of the hierarchical relationship. A concept may have any number of broader concepts: poly-hierarchies are allowed.

Example of using skos:broader/skos:narrower

```

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

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:prefLabel>Bangers and mash</skos:prefLabel>
    <skos:altLabel>Sausage and mash</skos:altLabel>
    <skos:altLabel>Sausage and mashed potato</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:broader rdf:resource="http://example.com/Concept/0002"/>
  </skos:Concept>

  <skos:Concept rdf:about="http://example.com/Concept/0002">
    <skos:prefLabel>English cuisine</skos:prefLabel>
    <skos:altLabel>English dishes</skos:altLabel>
    <skos:altLabel>English food</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:narrower rdf:resource="http://example.com/Concept/0001"/>
  </skos:Concept>

</rdf:RDF>

```

3.7.2. Related

skos:related property hierarchy

```

skos:semanticRelation
|
+--skos:related

```

The `skos:related` property allows you to create associative links between concepts. Again, this property carries weak semantics - it expresses the fact that two concepts are in some way related, and that the relationship should NOT be used to create a hierarchy. Essentially, this property allows you to create links between branches of a hierarchy of concepts.

Example of using `skos:related`

```

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

  <skos:Concept rdf:about="http://example.com/Concept/0002">
    <skos:prefLabel>English cuisine</skos:prefLabel>
    <skos:altLabel>English dishes</skos:altLabel>
    <skos:altLabel>English food</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:related rdf:resource="http://example.com/Concept/0003"/>
  </skos:Concept>

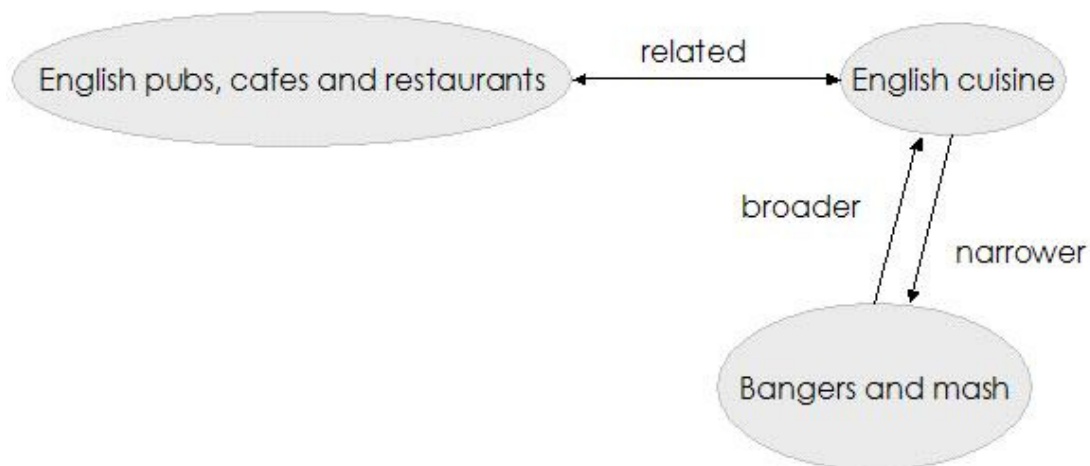
  <skos:Concept rdf:about="http://example.com/Concept/0003">
    <skos:prefLabel>English pubs, cafes and restaurants</skos:prefLabel>
    <skos:altLabel>English eateries</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:related rdf:resource="http://example.com/Concept/0002"/>
  </skos:Concept>

</rdf:RDF>

```

The two examples above give a concept scheme with the following structure:

Diagram of concept structure



3.8. Multilingual labelling [\[back to contents\]](#) -

A concept may be given labels from different languages. To specify the language of a label, use the language attribute of the RDF literal (see RDF specs). For each language, a concept may have one preferred label and any number of alternative labels. For example:

Example of multilingual labelling

```

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

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:prefLabel xml:lang="en">English cuisine</skos:prefLabel>
    <skos:altLabel xml:lang="en">English dishes</skos:altLabel>
    <skos:altLabel xml:lang="en">English food</skos:altLabel>
    <skos:prefLabel xml:lang="fr">Cuisine anglaise</skos:prefLabel>
    <skos:altLabel xml:lang="fr">Plats anglais</skos:altLabel>
    <skos:prefLabel xml:lang="es">Cocina ingles</skos:altLabel>
    <skos:prefLabel xml:lang="it">Cucina inglese</skos:prefLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:Concept>

</rdf:RDF>

```

3.9. More precise semantics [\[back to contents\]](#) -

The SKOS-Core property set is designed to be extensible. The `skos:semanticRelation` property is the top-level property for all properties linking concepts within the same concept scheme. The first level extension of this property is the three properties already mentioned, `skos:broader`, `skos:narrower` and `skos:related`. These may in turn be extended, to create properties that carry more precise semantics. This structure provides a framework for interoperability between concept schemes with different levels of precision in the semantics of their concept-to-concept relations.

A number of extensions are built in to the SKOS-Core 1.0 schema, and these are described below.

3.9.1. BroaderGeneric / NarrowerGeneric

`skos:broaderGeneric/skos:narrowerGeneric` property hierarchy

```

skos:semanticRelation
  |
  +--skos:broader
    |
    +--skos:broaderGeneric

skos:semanticRelation
  |
  +--skos:narrower
    |
    +--skos:narrowerGeneric

```

```

rdfs:subClassOf
|
+--skos:broaderGeneric

```

This extension of the `skos:broader/skos:narrower` property pair may only be used to specify a class subsumption relationship between two concepts. The semantics of these properties are inherited from the `rdfs:subClassOf` property.

Example of using `skos:broaderGeneric/skos:narrowerGeneric`

```

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

  <skos:Concept rdf:about="http://example.com/Concept/0004">
    <skos:prefLabel>English pubs, clubs and bars</skos:prefLabel>
    <skos:altLabel>English drinking establishments</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:narrowerGeneric rdf:resource="http://example.com/Concept/0005"/>
  </skos:Concept>

  <skos:Concept rdf:about="http://example.com/Concept/0005">
    <skos:prefLabel>English pubs</skos:prefLabel>
    <skos:altLabel>English public houses</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:broaderGeneric rdf:resource="http://example.com/Concept/0004"/>
  </skos:Concept>

</rdf:RDF>

```

3.9.2. BroaderInstantive / NarrowerInstantive

`skos:broaderInstantive/skos:narrowerInstantive` property hierarchy

```

skos:semanticRelation
|
+--skos:broader
|
+--skos:broaderInstantive

skos:semanticRelation
|
+--skos:narrower
|
+--skos:narrowerInstantive

rdf:type
|
+--skos:broaderInstantive

```

Use these properties to express an instantive (instance-of) relationship between two concepts. The semantics of these properties are inherited from the `rdf:type` property.

Example of using `skos:broaderInstantive/skos:narrowerInstantive`

```

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

  <skos:Concept rdf:about="http://example.com/Concept/0006">
    <skos:prefLabel>The Red Lion, Blewbury</skos:prefLabel>
    <skos:altLabel>The Red Lion pub in Blewbury</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:broaderInstantive rdf:resource="http://example.com/Concept/0005"/>
  </skos:Concept>

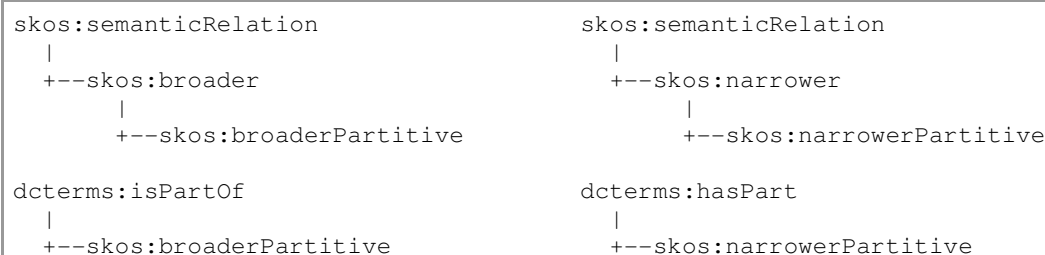
  <skos:Concept rdf:about="http://example.com/Concept/0005">
    <skos:prefLabel>English pubs</skos:prefLabel>
    <skos:altLabel>English public houses</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:narrowerInstantive rdf:resource="http://example.com/Concept/0006"/>
  </skos:Concept>

</rdf:RDF>

```

3.9.3. BroaderPartitive / NarrowerPartitive

skos:broaderPartitive/skos:narrowerPartitive property hierarchy



Use these properties to express a partitive (part-of) relationship between two concepts, where you would like such relationships to be organised as part of a hierarchical structure. The semantics of these properties are inherited from the `dcterms:isPartOf/dcterms:hasPart` property pair [see ref].

Example of using `skos:broaderPartitive/skos:narrowerPartitive`

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

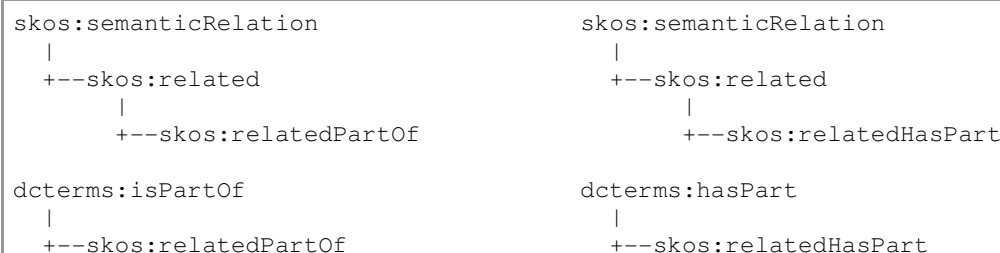
  <skos:Concept rdf:about="http://example.com/Concept/0007">
    <skos:prefLabel>Oxfordshire county</skos:prefLabel>
    <skos:altLabel>Oxon</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:broaderPartitive rdf:resource="http://example.com/Concept/0008"/>
  </skos:Concept>

  <skos:Concept rdf:about="http://example.com/Concept/0008">
    <skos:prefLabel>England</skos:prefLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:narrowerPartitive rdf:resource="http://example.com/Concept/0007"/>
  </skos:Concept>

</rdf:RDF>
```

3.9.4. RelatedPartOf / RelatedHasPart

skos:relatedPartOf/skos:relatedHasPart property hierarchy



Use these properties to express a partitive (part-of) relationship between two concepts, where you would like such relationships to be organised as part of an associative structure. These properties carry the same semantics as the `skos:broaderPartitive/skos:narrowerPartitive` property pair. These two options exist to provide a framework for interoperability between thesauri that treat the partitive relationship in different ways.

Example of using `skos:relatedPartOf/skos:relatedHasPart`

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

  <skos:Concept rdf:about="http://example.com/Concept/0001">
    <skos:prefLabel>Bangers and mash</skos:prefLabel>
    <skos:altLabel>Sausage and mash</skos:altLabel>
    <skos:altLabel>Sausage and mashed potato</skos:altLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:Concept>

</rdf:RDF>
```

```

        <skos:relatedHasPart rdf:resource="http://example.com/Concept/0009"
    </skos:Concept>

    <skos:Concept rdf:about="http://example.com/Concept/0009">
        <skos:prefLabel>Sausages</skos:prefLabel>
        <skos:altLabel>Bangers</skos:altLabel>
        <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
        <skos:relatedPartOf rdf:resource="http://example.com/Concept/0001"
    </skos:Concept>

</rdf:RDF>

```

3.10. High-level grouping of concepts [\[←back to contents\]](#) -

Many thesauri contain the notion of high-level grouping of concepts. This is done by arranging concepts in hierarchies under special concepts, these special concepts often referred to as 'fundamental facets' or 'fundamental categories'.

To represent the fact that a concept is intended to sit at the top of the concept hierarchy in a scheme, use the `skos:TopConcept` class.

skos:TopConcept class hierarchy

```

skos:Concept
|
+--skos:TopConcept

```

Example of defining a top concept

```

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

  <skos:TopConcept rdf:about="http://example.com/Concept/0010">
    <skos:prefLabel>Materials</skos:prefLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:TopConcept>

  <skos:TopConcept rdf:about="http://example.com/Concept/0012">
    <skos:prefLabel>Processes</skos:prefLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
  </skos:TopConcept>

</rdf:RDF>

```

Other concepts in the scheme may then be linked to a top concept via any of the standard semantic relation properties, e.g. `skos:broader` / `skos:narrower`. For example:

Example of linking concepts to top concepts

```

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

  <skos:TopConcept rdf:about="http://example.com/Concept/0010">
    <skos:prefLabel>Materials</skos:prefLabel>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:narrower rdf:resource="http://example.com/Concept/0011"/>
  </skos:TopConcept>

  <skos:Concept rdf:about="http://example.com/Concept/0011">
    <skos:prefLabel>Marble</skos:prefLabel>
    <skos:scopeNote>A granular crystalline limestone</skos:scopeNote>
    <skos:inScheme rdf:resource="http://example.com/thesaurus"/>
    <skos:broader rdf:resource="http://example.com/Concept/0010"/>
  </skos:Concept>

</rdf:RDF>

```

Of course, a top concept is not allowed to have any broader concepts.

4. Using SKOS-Core with Dublin Core for subject-based indexing [\[back to contents\]](#)

One of the main purposes of creating a concept scheme is to use it to help organise a set of resources e.g. a collection of web documents. When metadata about resources is enriched to include some statement about the 'subject' or 'topic' of the resource, and the 'subject' or 'topic' is a concept in some concept scheme, we can use that concept scheme to help search and browse around the collection of resources.

To use SKOS-Core with Dublin Core, make concepts the objects of statements using the `dc:subject` property. For example:

Example of using SKOS-Core with `dc:subject`

```
<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/">

  <rdf:Description rdf:about="http://example.com/web/aDocument.html">
    <dc:subject rdf:resource="http://example.com/Concept/0005"/>
  </rdf:Description>

</rdf:RDF>
```

Currently we recommend that concepts defined using SKOS-Core are NOT used with the `foaf:topic` or `foaf:primaryTopic` properties.

5. Further Reading [\[back to contents\]](#)

That's the end of this guide. Development of SKOS-Core, and the sister vocabulary SKOS-Mapping (which allows you to assert mappings between concepts from different schemes), is ongoing. Latest news and updates can be found on the website for the [SWAD-Europe Thesaurus Activity](#). You can also track the work via the public mailing list public-esw-thes@w3.org - all contributions are very welcome, and an [archive of previous mails](#) is also available.

Other reports and schemas from SWAD-Europe Thesaurus Activity -

[SWAD-Europe Deliverable 8.2: Review of Existing Work on RDF Thesaurus Schemas.](#)

[SWAD-Europe Deliverable 8.3: Multilingual Thesauri.](#)

[SWAD-Europe Deliverable 8.4: Inter-Thesaurus Mapping.](#)

[SKOS-Mapping: an RDF Schema for Inter-Thesaurus Mapping \(pre-release version\)](#)

Supporting and related guides -

[RDF Primer](#)

[OWL Web Ontology Language Guide](#)

[The FOAF Project](#)

[Expressing Qualified Dublin Core in RDF/XML](#)