Aalto University
School of Science

SeCo

SKOS
Simple Knowledge Organization System

Eerl Hyvönen

Aalto University, Semantic Computing Research Group (SeCo)
http://seco.cs.aalto.fi
firstname.lastname@aalto.fi

Learning objective

- Learn SKOS standard for representing simple KOSs, such as thesauri and classifications, as domain specific ontologies

Terminology: Vocabularies vs. Ontologies

Term ”Vocabulary” is often used in two meanings

1. Vocabulary = metadata model, by which a particular domain can be represented
- RDF Schema
- Dublin Core
- SKOS standard

2. Vocabulary = a particular Knowledge Organization System (KOS) for representing concepts of an application domain, such as a thesaurus
- ” Vocabularies” are created using ”vocabularies”?

Both (1) and (2) are often called “ontologies”

To disambiguate meanings, I will use term “domain ontology” to refer to KOSs
- Keyword thesauri
- Place gazetteers
- Authority file systems (for persons, organizations)

SKOS Standard: An Introduction

Aalto University
School of Science

Department of
Computer Science

SeCo

Semantic Web Standards of W3C for Representing Ontologies

- RDF Schema
  - Class hierarchies, property hierarchies, instances, constraints
- **SKOS Simple Knowledge Organization System**
  - **for representing simple domain ontologies**
- OWL Web Ontology Language
  - for ontologies with richer semantics

SKOS Overview

- RDF-based standard vocabulary for representing KOSs as ontologies:
  - *thesauri, classification schemes, subject heading systems , taxonomies etc.*
- Focus: represent KOS structures rather than real world knowledge
  - *In SKOS: entries, e.g., ”table” are instances of skos:Concept*
  - *In RDF(S): entries, e.g., ”table” are classes of from which actual instances (of tables) are created*
- SKOS can be extended using RDF(S) and OWL

Major Domain Ontology Types

General concepts
- Used, e.g., as subject keywords, representing object types
- E.g., chair, studying, philosophy, …

Actors
- Persons, groups, and organizations
- Individuals with rich metadata
- E.g., Napoleon I, impressionists, Nokia Corp.

Places
- Points, paths, areas with geolocation
- Individuals with rich metadata
- Senate Square, Ring II, Finland

Times

Events
- Birth of Jesus, World War II
- Events are “semantic glue” thank link domain ontologies: actors, places, and times

SKOS vocabulary specification

SKOS Simple Knowledge Organization System Reference

W3C Recommendation 18 August 2009

This version:
http://www.w3.org/TR/2009/REC-skos-reference-20090818/
Latest version:
http://www.w3.org/TR/skos-reference
Previous versions:
http://www.w3.org/TR/2009/PR-skos-reference-20090615/
Editors:
Alistair Miles, STFC Rutherford Appleton Laboratory / University of Oxford
Sean Bechhofer, University of Manchester

Please refer to the **errata** for this document, which may include some normative corrections.

See also **translations**.

In SKOS Ontologies are Concept Schemes

Ontologies are instances of concept schemes

- Top concepts are pointed out

```
ex:animalThesaurus rdf:type skos:ConceptScheme ;
    dct:title "Simple animal thesaurus" ;
    dct:creator ex:antoineIsaac ;
    skos:hasTopConcept ex:mammals ;
    skos:hasTopConcept ex:fish .
```

dct: <http://purl.org/dc/terms/> (Dublin Core terms)

Concepts and Labels

Concepts (for machines)

- Instances of skos:Concept
- **`ex:animals rdf:type skos:Concept .`**

Labels (human-readable names of concepts)

- skos:prefLabel, skos:altLabel, skos:hiddenLabel
- For example:
  - ***`ex:animals rdf:type skos:Concept ;`***
    `skos:prefLabel "animals"@en ;`
    `skos:altLabel "creatures"@en ;`
    `skos:prefLabel "animaux"@fr ;`
    `skos:altLabel "créatures"@fr .`

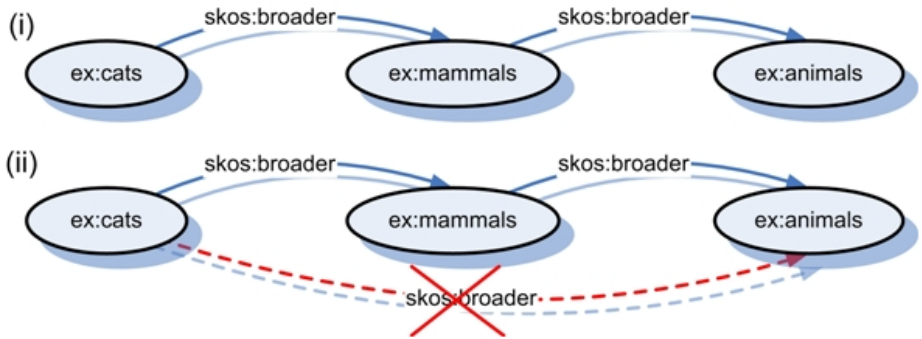
Documentary Notes

- skos:note
  - *skos:scopeNote*
  - *skos:editorialNote*
  - *skos:changeNote*
  - *skos:historyNote*
- skos:definition
- skos:example

Semantic Relationships

Broader/Narrower relationships

- Like in thesauri
-
```
ex:animals rdf:type skos:Concept ;
           skos:prefLabel "animals"@en ;
           skos:narrower ex:mammals .
```
- skos:narrower / skos:broader are ***not*** transitive!



Example:
Thesaurus Limitations for Term Expansion

- Furniture
  NT Mirrors
  Mirrors
  NT Makeup mirrors

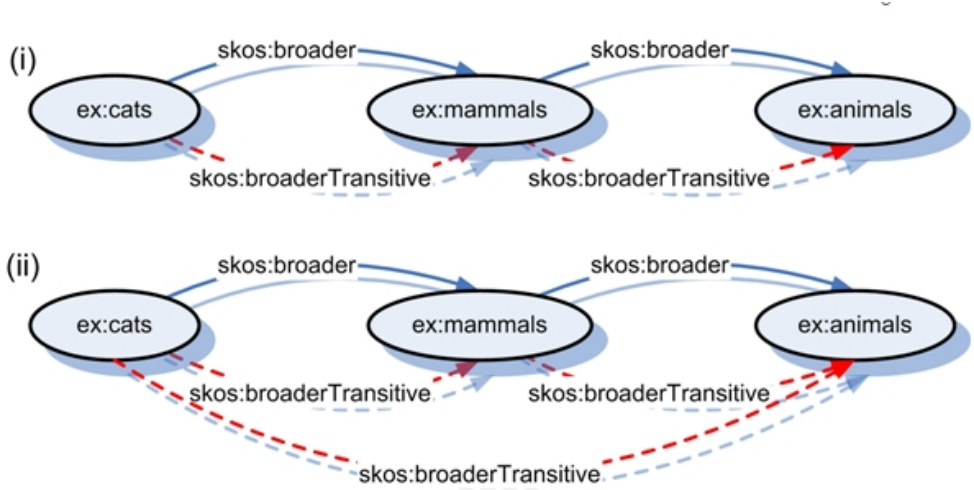
- OK, but the results of query ”Find all furniture”
  would contain also makeup mirrors in term expansion!

- Thesaurus hierarchies are not necessarily transitive!

Modeling Transitivity

skos:broaderTransitive and skos:narrowerTransitive

- skos:broader / skos:narrower are their subproperties
- Transitive versions are inferred, not asserted



Other Semantic Relationships

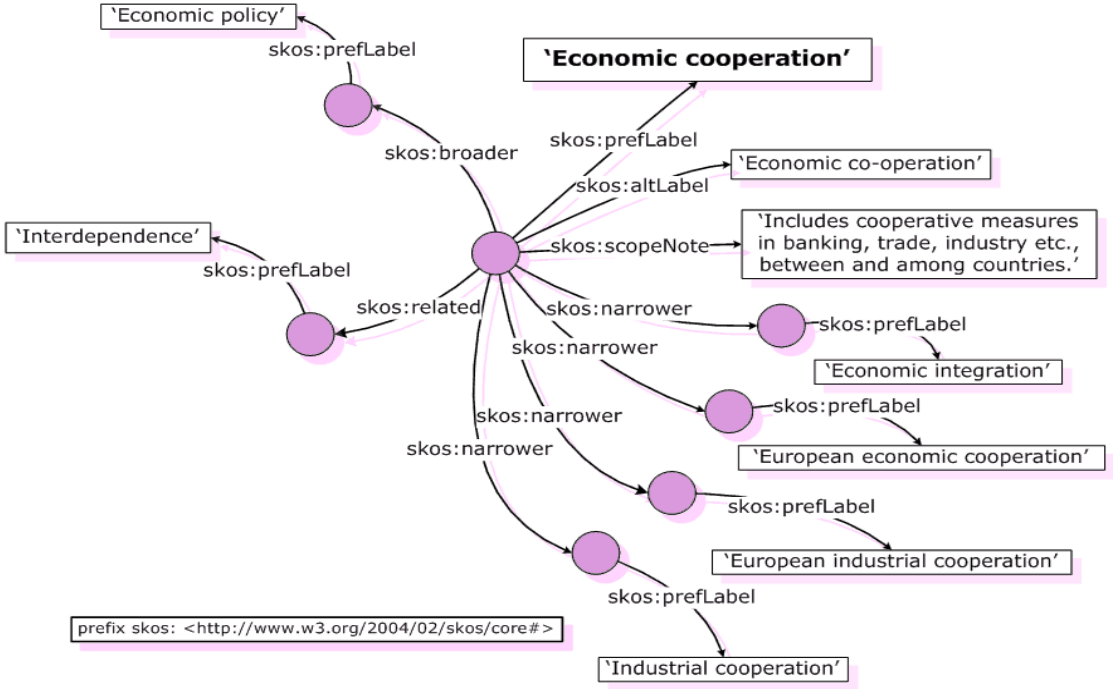
Associative Relationships

-
```
ex:birds rdf:type skos:Concept ;
         skos:prefLabel "birds"@en ;
         skos:related ex:ornithology .
```

skos:broader/narrower/related properties can be refined by introducing more specific properties using rdfs:subPropertyOf

-
```
ex:partOf rdfs:subPropertyOf skos:broader .
```

SKOS Ontology Example



(SKOS Core Guide, 2005)

Mapping Concept Schemes

- Properties for aligning concepts **between different ontologies**
- Equality (between two concept schemes)
  - *skos:exactMatch*
  - *skos:closeMatch*
- Other semantic relations (subproperties of concept properties)
  - *skos:broadMatch (< skos:broader)*
  - *skos:narrowMatch (< skos:narrower)*
  - *skos:relatedMatch (< skos:related)*

Other features

skos:Collection

- *For grouping concepts by a criterion. Collections are **not** used for indexing and can be blank nodes:*

```
ex:milk rdf:type skos:Concept ;
                skos:prefLabel "milk"@en .
ex:cowMilk rdf:type skos:Concept ;
                    skos:prefLabel "cow milk"@en ;
                    skos:broader ex:milk .
ex:goatMilk rdf:type skos:Concept ;
                     skos:prefLabel "goat milk"@en ;
                     skos:broader ex:milk .
_:b0 rdf:type skos:Collection ;
          skos:prefLabel "milk by source animal"@en;
          skos:member ex:cowMilk ;
          skos:member ex:goatMilk .
```

milk
<milk by source animal>
cow milk
goat milk
<milk by fat level>
…

(van Assem, Isaac, 2005)

Other Features (2)

skos:OrderedCollection (class)

- Same idea as with collections, but members are represented as an ordered list (instance of class rdf:List)
  - *Ordering information of subconcepts is often needed in, e.g., user interfaces, and collections do not represent it*

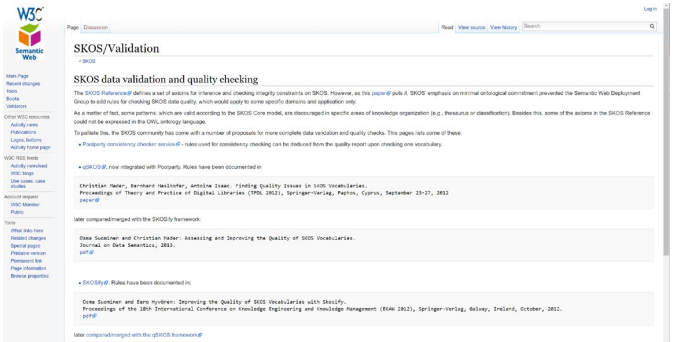
skos:notation (property)

- String of characters/code used to uniquely identify a concept within a concept scheme, based on the original KOS identifiers
  - `ex:semanticWebCourse skos:notation "CS-E4410" .`

SKOS Inference and Validation

SKOS Reference defines also:

- **set of axioms for inference for enriching data**
  - E.g., reasoning skos:broader/narrowerTransitive triples
- **set of integrity conditions for validating SKOS ontologies**
  - E.g., the hierarchies should not have loops
  - https://www.w3.org/2001/sw/wiki/SKOS/Validation



SKOS-XL:
Extension for Richer Label Modeling

Modeling labels as resources: SKOS-XL

- Labels are instances of skosxl:Label
  - *Then labels can have properties, too*
- Label instances are referred to using
  - *skosxl:prefLabel*
  - *skosxl:altLabel*
  - *skosxl:hiddenLabel*

SKOS-XL Example

```
<Love>
  skosxl:prefLabel <A> ;
  skosxl:altLabel <B> ;
  skosxl:hiddenLabel <C> .

<A> rdf:type skosxl:Label ;
         skosxl:literalForm "love"@en ;
         dct:created "2006-10-03"^^xsd:date .

<B> rdf:type skosxl:Label ;
         skosxl:literalForm "adoration"@en ;
         dct:created "2006-10-03"^^xsd:date .

<C> rdf:type skosxl:Label ;
         skosxl:literalForm "luv"@en ;
         dct:created "2015-05-14"^^xsd:date .
```

More Information

Namespace IRIs of SKOS and SKOS-XL contain the specifications in RDF for 1) classes and 2) properties

- http://www.w3.org/2004/02/skos/core#
  - *Direct link to RDF serialization:* http://www.w3.org/2004/02/skos/core.rdf
- http://www.w3.org/2008/05/skos-xl#
  - *Direct link to RDF serialization:* http://www.w3.org/2008/05/skos-xl.rdf

SKOS home page and documentations for a full list of features

For more elaborate ontology modeling there is the OWL standard
(topic of the next lecture)

Why Use Ontologies?

- Semantic interoperability based on sharing concepts
- Benefits of reasoning
  - *Ontology development*
    - Deriving implicit subsumption hierarchies
    - Consistency checking is possible
    - Ontology matching and merging
  - *Information retrieval*
    - Query expansion
    - Reasoning: finding implicit results based on reasoning
      - *Semantic search methods and recommending*
    - Using ontology structures to aid query formation and presenting results
    - Semantic analysis of contents

Questions