Università di Roma Tor Vergata

University of Rome “Tor Vergata”

SKOS
Simple Knowledge Organization System

Armando Stellato

stellato@uniroma2.it

lastupdate: Fri, 11 May 2012 14:56:00 +000

most of this material has been contributed by Manuel Fiorelli

Introduction (1)

A **collection** of books, papers, artworks, ... **is useful** only if there are means to:

- *understand the structure* of the collection;
- *efficiently retrieve things* which users are interested in;
- *navigate* through the collection;
- *discover new things* in the collection.

Librarians, curators, ... have met those requirements by means of **cataloguing systems**.

Introduction (2)

Semantic Web is shifting from the *époque* of "*ontologies everywhere*" to the *era of linked-data*:

- the main goal is to *make data available* on the web;
- *regardless the precise* definition of their *semantics*.

There always be **something** that can **not be described formally** through an ontology (e.g. documents, music, etc...)

But it is desirable to link those resources to other data available

- SKOS satisfies this need providing means for *indexing* resources with respect to a *weakly defined conceptualization*

What is SKOS?

Simple Knowledge Organization System (**SKOS**) is an **RDF vocabulary** (i.e. a set of RDF URI Reference) for describing a *Knowledge Organization System* (**KOS**).

In other words, SKOS is a **data-model** for representing KOSs.

SKOS is a **W3C Recommendation**.

The intended use of that vocabulary is specified as an **OWL Ontology**, although it is not sufficient to express every constraint precisely.

Knowledge Organisation Systems (1)

"The term **knowledge organization systems** is intended to encompass all types of schemes for *organizing information* and promoting *knowledge management*." (Hodge 2000)

The term was coined by the **Networked Knowledge Organization Systems Working Group** at its initial meeting at the *ACM Digital Libraries '98 Conference* in Pittsburgh, Pennsylvania.

Knowledge Organisation Systems (2)

The term **KOS may refer to**

- *classification* and *categorization schemes*
- *subject headings*
- *authority files*
- *thesauri*

- *semantic networks*
- *ontologies*

KOSs have grown in the field of *Library and Information Science*

- to organize *physical libraries*
- later, to organize *digital libraries*

They have been applied to several NLP tasks, IR, ...

Knowledge Organisation Systems (3)

A **classification scheme** organizes material at a *general level*.

It relates to the need of giving to each book a *single location on the shelf*.

The *Library of Congress Classification* is used in several libraries in the USA and other countries.

Class A - *General Works*
  Subclass **AC** - Collections. Series. Collected works
  Subclass **AE** - Encyclopedias
  ...
Class B - *Philosophy, Psychology and Religion*
  Subclass **B** - *Philosophy* (General)
  Subclass **BC** - *Logic*
  ...
...

Knowledge Organisation Systems (4)

A collection of **subject headings** reflects a *more detailed* organization of the material.

A given entity (e.g. a book, a chapter, ...) may be provided with *multiple* subject headings.

E.g.

Library of Congress Subject Headings

Medical Subject Headings (MeSH)

The *Medical Subject* Headings comprise the U.S. National Library of Medicine's controlled vocabulary used for indexing articles, for cataloging books, ...

Knowledge Organisation Systems (5)

An **authority file** controls the *variant names* for an entity of the domain value for a particular field.

E.g.

Library of Congress Name Authority File

Those files generally have a rather *flat structure*.

Whaat is SKOS for? (1)

Many kinds of KOSs (*thesauri, taxonomies, classification schemes* and *subject heading systems)* have arisen in *different applications* and *domains*.

SKOS provides a *fast path* for the *migration* of existing resources to the **Semantic Web**.

That should help to ***share*** and ***link*** KOSs through the Web.

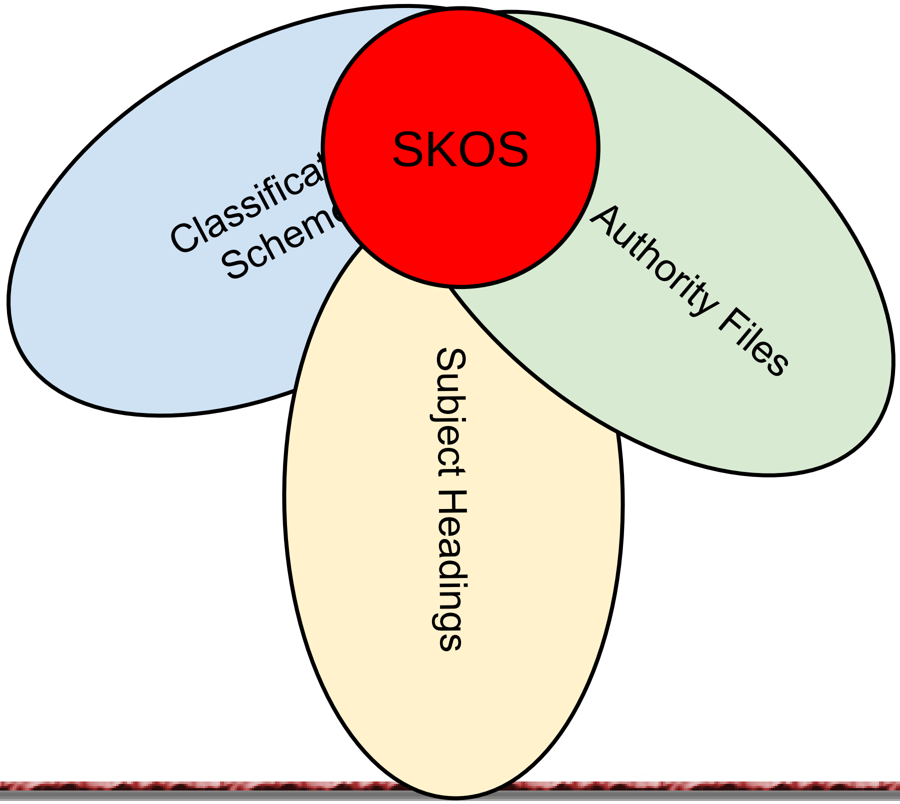
The adoption of a standard data model entails a vast **technology reuse**: e.g. state-of-the-art *triple stores* for the data management, *SPARQL* as a query language, Turtle or RDF/XML as a serialization format.

Armando Stellato stellato@uniroma2.it

What is SKOS for? (2)

There are **several kinds of KOSs**.

How can they be **mapped** to a **single standard data-model**?



What isn't SKOS for?

SKOS is **not for replacing** *existing KOSs* in the applications where they have been developed.

SKOS **doesn't aim to replace** existing guidelines for the compilation of KOSs.

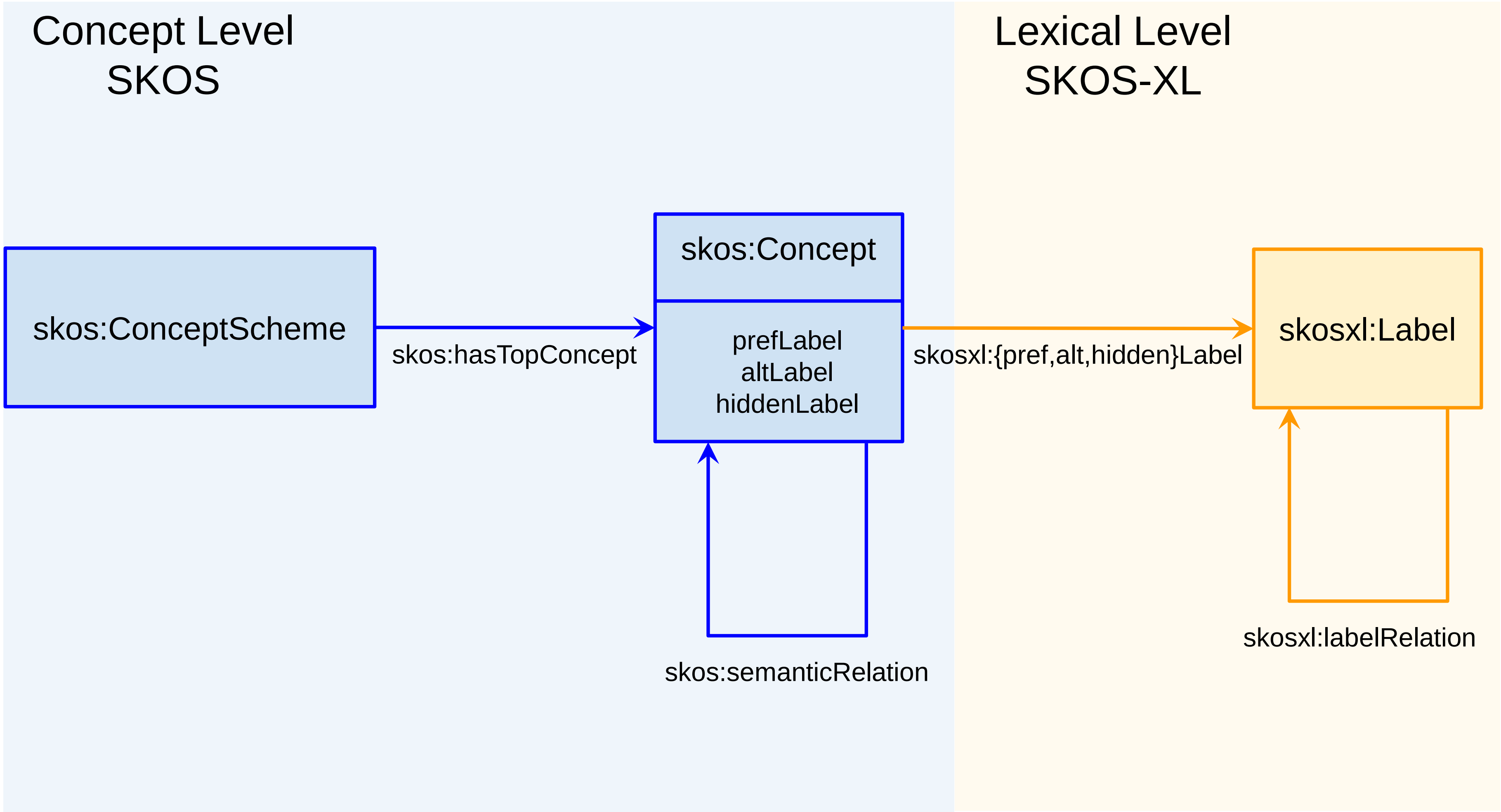
SKOS in short (1)

SKOS models the **least common denominator** across several kinds of KOSs.

According to **SKOS-REFERENCE**:

"Using SKOS, **concepts** can be identified using URIs, **labeled** with lexical strings in one or more natural languages, assigned **notations** (lexical codes), **documented** with various types of note, **linked to other concepts** and organized into informal hierarchies and association networks, aggregated into **concept schemes**, grouped into labeled and/or ordered **collections**, and **mapped** to concepts in other schemes."

SKOS in short (2)



Armando Stellato stellato@uniroma2.it
http://art.uniroma2.it/stellato

SKOS in short (3)

The scheme below is not specified by SKOS.

SKOS allows for the definition of **concept-based** KOSs.

Conceptual level

Concepts represents the different *senses* born by lexical items.

Semantic **relations between** concepts represents precisely hierarchical, associative and other kind of *connections* which do *not dependent on lexicalization*.

Terminological correspondence

Concepts are associated with their (preferred, alternative, ...) lexicalizations.

Lexical level

Lexical relations represent connections between different *lexical items*. (only in **SKOS-XL**) The nature of a lexical relation is up to SKOS-XL users.

SKOS versus OWL (1)

The Semantic Web has already a rich modelling language, named OWL.

What it the need of another language?

They have been introduced for **different purposes**:

- **OWL** is a formal *knowledge representation* language;
- **SKOS** is a language for the definition of *simple* conceptualizations, mainly targeted to **IR applications**.

SKOS versus OWL (2)

In an OWL ontology **classes** represent **groups of individuals** sharing properties.

OWL provides a **set of terms** for:

- defining *classes*;
- defining *properties*.

```
ex:Person rdf:type owl:Class .
ex:name rdf:type owl:DatatypeProperty ;
        rdfs:domain ex:Person ;
        rdfs:range xsd:string .
```

Usually we apply those properties to individuals.

```
ex:manuel ex:name "Armando Stellato"^^xsd:string .
```

SKOS versus OWL (3)

Usually we deal with a *syntactic subset* of OWL named **OWL-DL** corresponding to a *decidable** language within the family of Description Logics.

*most reasoning tasks are guaranteed to be solvable by an always terminating procedure

OWL-DL mandates the **separation** among classes (which may be thought as binary predicates), *individuals* (which may be thought as monadic predicates) and *data values*.

The constraint above *mostly* **prevent to predicate over classes** (beyond the terms provided by OWL).

SKOS versus OWL (4)

Most KOSs do not require the distinction between classes and individuals, either because:

- there are no individuals at all,
- the relation between individuals and classes is not expected to produces particular inferences.

Hence, there is no need to treat concepts (in the sense of SKOS) as classes, but it is sufficient to treat them as individuals of the class `skos:Concept`.

SKOS versus OWL (5)

SKOS and OWL are **subtly related**, since SKOS may be seen as an OWL vocabulary.

Hence, a SKOS description is in fact an **OWL ontology**.

That opens it up to advanced modelling solutions, where SKOS and OWL constructs are **interwoven**.

To fix the ideas it is worth summarizing what OWL is for, and how SKOS fits with that architecture.

Armando Stellato stellato@uniroma2.it
http://art.uniroma2.it/stellato

OWL - in short (1)

Web Ontology Language (**OWL**) has been introduced to allow the *formal specification of vocabularies* with a level of expressiveness beyond what was offered by **RDFS**.

In OWL you have not to declare a-priori what a resource is used for: individual, class, meta-class; however, it is useful to think about resources in terms of three levels: M-0, M-1, M-2 (borrowed from *Model Driven Engineering*).

In OWL-DL that separation *exists de-facto*.

OWL - in short (2)

At **M-0**: we *assert facts about individual*

```
ex:bob rdf:type ex:Person .
ex:bob ex:wife ex:susan .
```

but what about the terms `ex:wife` and `ex:Person`?

At **M-1**: we *define a conceptualization* --> we *predicate about the vocabulary*

```
ex:Person rdf:type owl:Class .
ex:wife rdf:type owl:ObjectProperty ;
        rdfs:domain ex:Person ;
        rdfs:range ex:Person .
```

At **M-2**: we have the vocabulary definition language (e.g. OWL)

the definition of OWL is built-in into the reasoner

OWL Interpretation of SKOS (1)

At **M-1**: SKOS is defined as an OWL vocabulary

```
skos:Concept rdf:type owl:Class .
skos:semanticRelation rdf:type owl:ObjectProperty ;
                               rdfs:domain skos:Concept ;
                               rdfs:range skos:Concept .
[...]
```

At **M-0**: a KOS is represented by means of the SKOS vocabulary

```
ex:animals rdf:type skos:Concept
```

SKOS defines a *conceptualization over KOSs*, thus providing a vocabulary for specifying conceptualizations over domains.

A given KOS (e.g. the Library of Congress Classification) may be seen an instance of the SKOS model.

OWL Interpretation of SKOS (2)

SKOS covers deliberately only the least common denominator among several kinds of KOSs.

Missing information may be represented by means of a ***dedicated vocabulary***, which has to be defined (at level M-1), possibly by **specialization of** the **SKOS** vocabulary.

For example, lexical property `ex:acronymOf` can be defined:

```
ex:acronymOf rdfs:subPropertyOf skos:labelRelation .
```

SKOS in short - concepts (1)

A **concept** is a "*unit of thought*" (i.e. an idea, a meaning, a category of things, ...). It is the **fundamental unit** of every KOS. It is a "*suggestive rather than a restrictive*" definition.

Synsets, synonymous rings are candidate to be treated as concepts.

In SKOS concepts are modelled by the class `skos:Concept`.

Every concept is assigned a **URI**, which is used for *identification* purposes (even in different concept schemes and SKOS descriptions). According to the **Linked Data paradigm** that URI should be *dereferenceable* (not a SKOS requirement).

A concept is further characterised in terms of *labels*, *documentary notes*, notations and *semantic relations*.

A glimpse of Linked Data (1)

Linked Data is a paradigm for *publishing data on the web*.

Linked Data *relies on the web architecture* to create a web of data, analogous to the current web of documents.

The paradigm consists of *four principles*:

- identify resources with **URI**
- make those URI **dereferenceable** through HTTP
- **describe** resources in a **standard way** (e.g. use RDF)
- **embed links** to other resources within the description of a resource

A glimpse of Linked Data (2)

Web of documents	**Web of data**
information resources	entities (real-world entities, abstract entities, ...)
http://art.uniroma2.it/stellato (my academic home page)	http://data.art.uniroma2.it/stellato (an URI which might identify myself)
HTML	RDF
hypertextual link • links are untyped (except for the attribute *rel*) • but anchors convey a lot of information about the linked document	triples whose object denotes another entity (described in another place) • links are typed (by the predicate URI)

A glimpse of Linked Data (e.g. #1)

An unknown entity is identified by the following

http://aims.fao.org/aos/agrovoc/c_12332

You can dereference it and retrieve its RDF description

```
<http://aims.fao.org/aos/agrovoc/c_12332> a skos:Concept
<http://aims.fao.org/aos/agrovoc/c_12332> skos:prefLabel "maize"@en
<http://aims.fao.org/aos/agrovoc/c_12332> skos:narrower ↲
     <http://aims.fao.org/aos/agrovoc/c_1474>
```

The **discovery process** may continue until no new URI are found.

SKOS in short - labels (1)

*= ***not formally stated*** = a constraint which is mandated by the specs but not specified formally trough OWL

A label is an **expression** which is used in a **natural language** to refer to a concept.

Three (`owl:AnnotationProperty`) **properties**: (*disjoint, *plain-literal values, refine rdfs:label)

- **`skos:prefLabel`** the *preferred lexicalization* - *at *most one* in a given natural language - *unique* across a concept scheme for a given natural language (*best practice*)
- **`skos:altLabel`** *alternative expressions* (e.g. near-synonyms, abbreviations, acronyms) - *upward posting* is supported but *discouraged - one can refine it (e.g. acronym)*
- **`skos:hiddenLabel`** expressions which are provided only for *indexing purposes* (e.g. mispellings, stems, ...)

SKOS in short - labels (e.g. #1)

```
ex:animals     skos:prefLabel         "animals"@en ;
               skos:altLabel          "creatures"@en ;
               skos:prefLabel         "animali"@it  ;
               skos:altLabel          "creature"@it .
```

```
ex:fao rdf:type skos:Concept;

       skos:prefLabel "Food and Agriculture Organization"@en ;

       skos:altLabel "FAO"@en.
```

This labels are clearly related but that relationship cannot be represented (see later...)

SKOS in short - multilingual KOSs

A concept may be provided with lexicalizations in several languages.

```
ex:dog skos:prefLabel "cane"@it
ex:dog skos:prefLabel "dog"@en
ex:dog skos:prefLabel "狗"@zh-Hans
```
(phonetic "Gǒu")

UNICODE lexical forms handles any language.

Language tag are applied at literal level (*allowing for fine-grained localization*).

A glimpse of SKOS-XL (1)

In RDF[1] it is not possible to predicate about labels (e.g. relating them to each other), because they are *plain literals*, which aren't allowed to be the subject of a triple.

SKOS-XL (*eXtension for Labels*) is an extension of SKOS, which treats labels as first-class citizens.

The class **`skosxl:Label`** is introduced to model literals as individuals (in the OWL sense).

A **`skosxl:Label`** can be associated with a plain literal through the property `skosxl:literalForm`.

[1] also ER distinguishes between relations (among entities) and attributes of entities.

A glimpse of SKOS-XL (2)

A *concept may be **associated with** an XLabel* by means of one of the following properties:

- `skosxl:prefLabel`
- `skosxl:altLabel`
- `skosxl:hiddenLabel`

which *mirror the literal-based* labelling constructs.

Actually, the domain of that properties is not restricted to any class, thus they are applicable to any individual.

SKOS in short - labels (e.g. #2)

```
ex:acronymOf rdfs:subPropertyOf skosxl:labelRelation .

ex:fao rdf:type skos:Concept ;
       skosxl:prefLabel ex:label1 ;
       skosxl:altLabel ex:label2 .

ex:label1 rdf:type skosxl:Label ;
         skosxl:literalForm "Food and Agriculture ↲
  Organization"@en .

ex:label2 skos::type skosxl:Label ;
         skosxl:literalForm "FAO"@en .

ex:label2 ex:acronymOf ex:label1 .
```

SKOS in short - labels (3)

That is a compatible extension because the **property chain** `(skosxl:xxxLabel, skosxl:literalForm)` is a sub-property of `skos:xxxLabel`.

C skos<u>xl</u>:*xxx*Label XL
XL skosxl::literalForm **Label**

entails

C skos:*xxx*Label **Label**

An application may *safely ignore the SKOS-XL extension*, as long as a reasoner is able to produce the entailed triples.

SKOS in short - labels (e.g. #3)

```
ex:dog rdf:type skos:Concept ;
       skosxl:prefLabel ex:label3 .

ex:label3 rdf:type skosxl:Literal ;
          skosxl:literalForm "dog"@en .
```

The formal semantics assure that the following triple holds:

```
ex:dog skos:prefLabel "dog"@en .
```

SKOS in short - documentary notes

A **documentary note** provides an insight on the *meaning of a concept* or keeps track of the *editorial changes* within a KOS.

SKOS provides an *annotation property* named **`skos:note`**, which is further specialised into `skos:scopeNote`, `skos:definition`, `skos:example`, and `skos:historyNote`.

The existence of a super-property enables us to **collect** every documentary note.

SKOS in short - notations (1)

A **notation** is a string of characters that *uniquely identifies* a concept within a concept scheme.

```
skos:notation a owl:DatatypeProperty
```

This property has been introduced by retain a connection with *pre-existing classification schemes*.

A notation is by convention a **typed literal**, the datatype of which identifies the notation in use.

SKOS in short - notations (2)

Notations, *preferred labels* and *URIs* **seems to be overlapping** notions, but it is not the case.

Notations and *preferred labels* are only **assumed** to be **unique** within a given **concept scheme**: they are not **globally unique** identifier in contrast to *URIs*.

Also, a notation is *interpretable* unambiguously *only if the datatype is given* (e.g. the notation K may denote either the chemical element potassium or a black cartridge).

A **preferred label** is assumed to be in a *natural language*, whereas a **notation** generally refers to an *artificial classification notation.*

SKOS in short - semantic relations (1)

Semantic relations connect concepts together to create a semantic network.

Three **properties** (have `skos:Concept` as range and domain)

- `skos:broader` / `skos:narrower` they *should be read*: x HAS broader/narrower concept y - they map *hierarchical* taxonomic and aggregation relations - each is the *inverse* of the other - *non transitive* - *may be employed in reflexive statements (even if most KOSs forbid that)*
- `skos:related` *semantic (non hierarchical) associations* - *symmetric* - *non transitive*

SKOS in short - semantic relations (2)

Two concepts related by the transitive closure of `skos:broader` or `skos:narrower` cannot be connected with `skos:related`.

(not formally stated)

The three properties are not transitive, to avoid unexpected results due to the weak semantics of those properties.

SKOS in short - semantic relations (3)

It seems advantageous to have both `skos:narrower` and `skos:broader` (in contrast to OWL which only has `rdfs:subClassOf`).

But actually it may turn into a **disadvantage**.

- KOS consumers become dependant on the availability of a *reasoner* which materializes the *implicit relationships*
- but reasoners are often turned off for the sake of efficiency
- thus they have to *implement the procedure by hand*

If you have a relation oriented querying mechanism (eg. SPARQL) you don't need a new symbol for the inverse relation, but you have for free flipping the placeholders (in OWL-2 the notion of property^{-1} addresses this issue).

```
SELECT ?b ?n WHERE { ?n skos:broader ?b. }
```

SKOS in short - semantic relations (3b)

It is advisable to decide whether use `skos:narrower` of `skos:broader` and stick to that convention.

SKOS in short - concept schemes (1)

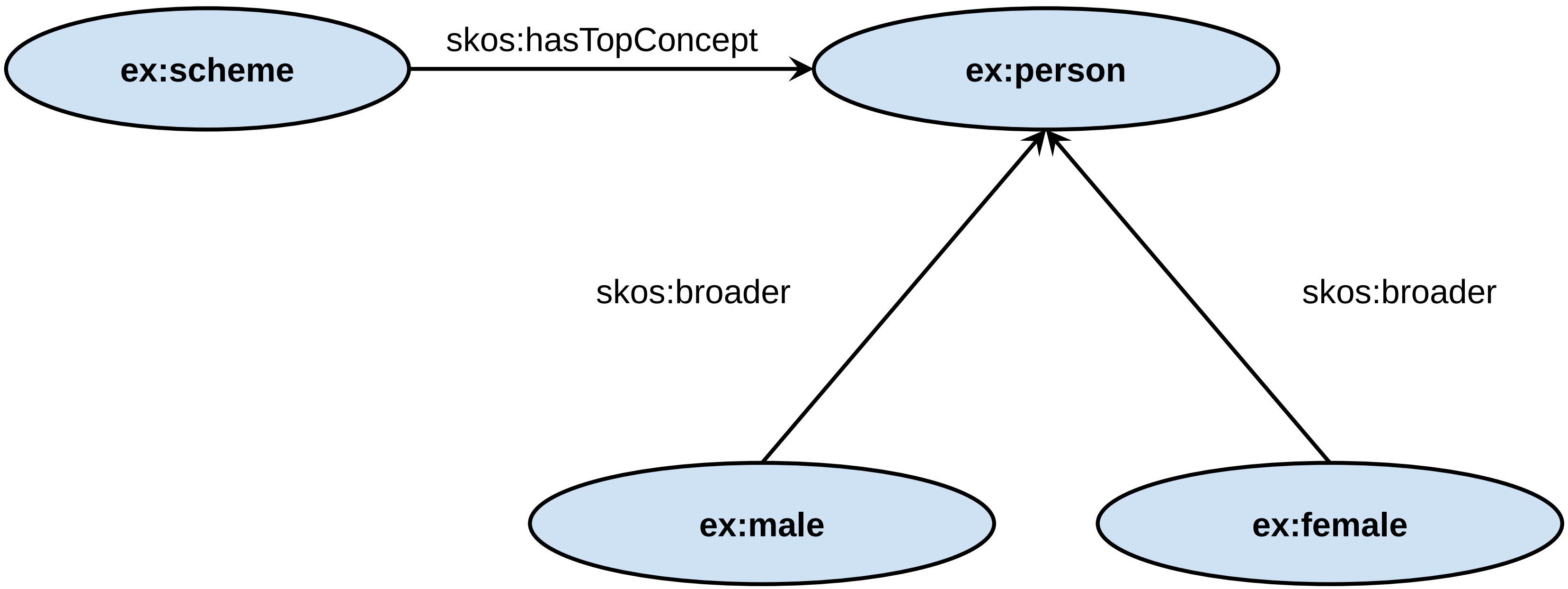
Concepts may be *grouped together* (via `skos:inScheme`) in **concepts schemes** (`skos:ConceptScheme`).

One would use a concept scheme, when he needs to *reify the* KOS in order to **attach metadata** to it.

Also, it supports the **coexistence** of several KOSs within a single RDF description, even if it *does not allow for recording* which statements about a concept pertain to a given concept scheme (*traditionally a KOS is made of concepts and informations about them*): e.g. getting all the narrower concepts of a given concept C requires an application to first list the narrower concepts of C and then filter out those not belonging to the scheme which is being browsed.

SKOS in short - concept schemes (2)

The property `skos:hasTopConcept` relates a *concept scheme* with a *concept*, which is **assumed** to be one of the *greatest elements* with respect to the *partial order induced by the taxonomic relation*, even if it is NOT required to be so.



SKOS in short - concept schemes (3)

S `skos:hasTopConcept` C **entails** C `skos:topConceptOf` S

C `skos:topConceptOf` S **entails** C `skos:inScheme` S

In a Linked Data context, the property `skos:hasTopConcept` provides a **set of anchors** to *start the navigation* of the concept scheme, without the need of knowing the whole hierarchy.

Advanced SKOS

- (Ordered) **collections** of Concepts (*not discussed here!*)
- **transitive properties** (*not discussed here!*)
- **mappings** between concepts

Advanced SKOS - Concept Mapping

```
skos:mappingRelation
  |- skos:narrowerMatch rdfs:subPropertyOf skos:narrower
  |- skos:broaderMatch  rdfs:subPropertyOf skos:broader
  |- skos:relatedMatch  rdfs:subPropertyOf skos:related
  |- skos:closeMatch a owl:SymmetricProperty
       |- skos:exactMatch a owl:SymmetricProperty ,
                            owl:TransitiveProperty
```

`x skos:closeMatch y` means that those concepts are sufficiently similar that they can be **used interchangeably** in **some** IR contexts.

`x skos:exactMatch y` means that those concepts are sufficiently similar that they can be **used interchangeably** in **most** IR contexts.

x and y remain *distinct individual* with their own properties (e.g. preferred labels), in contrast to what would have happened with `owl:sameAs`.

SKOS in action - Information Retrieval

OWL classes serve to model **shared characteristics** among a group of individual, enabling inference.

SKOS concepts (instead) serve solely the purpose of providing **indexing terms** for organizing resources (without worrying about formal definitions and inferences).

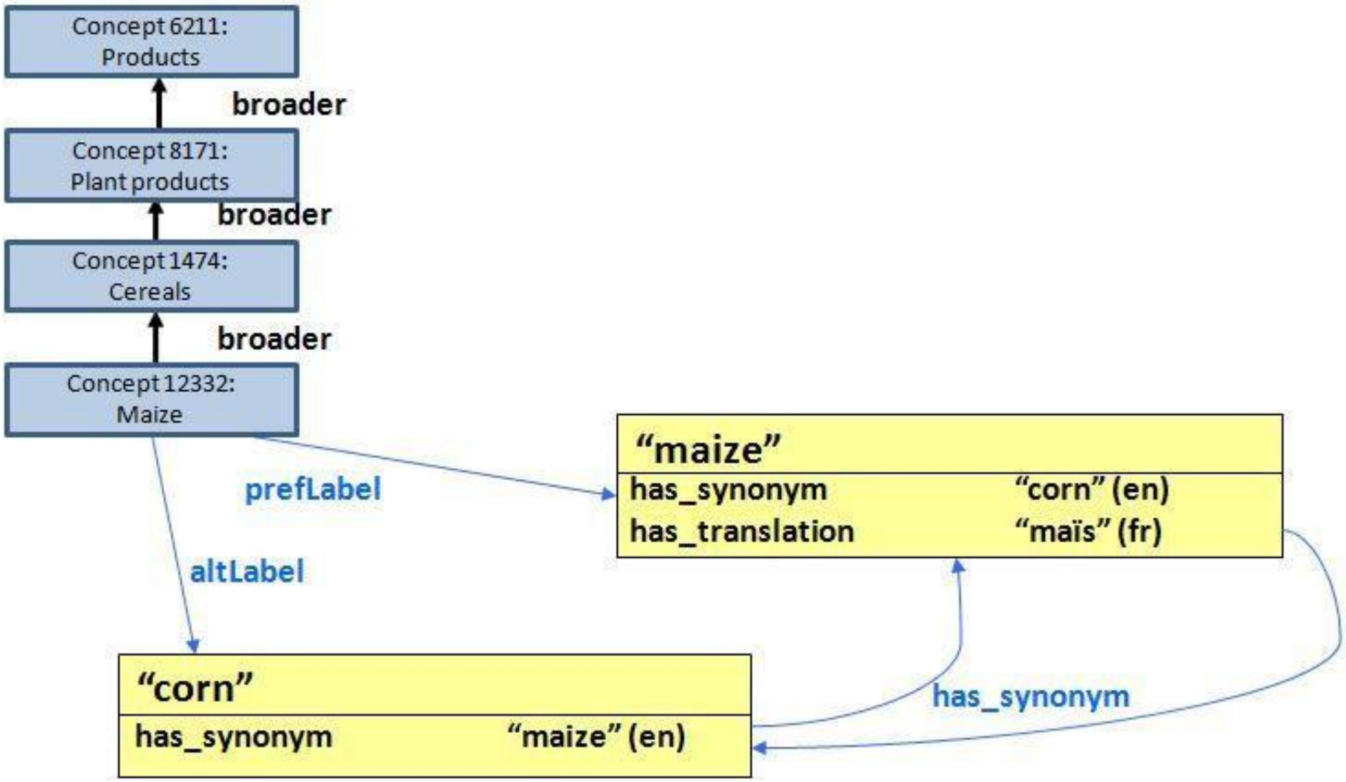
<u>A SKOS concept generally denotes a set of focussed documents</u>.

A SKOS description (and KOSs in general) may be used:

- ***transparently***: to perform **query expasion**
- ***explicitly***: to power the **navigation structure** of a repository

SKOS in action - AGROVOC

The **AGROVOC** *thesaurus* (compiled by FAO) contains more than **30 000 concepts** in up to **21 languages** covering topics related to *food*, *nutrition*, *agriculture*, *fisheries*, *forestry*, *environment* and other related domains.



SKOS in action - EuroVoc

EuroVoc is a *multilingual*, *multidisciplinary thesaurus* covering the *activities of the EU*, the European Parliament in particular. It contains terms in 22 EU languages ([...]), plus Croatian and Serbian.

SKOS in action - Data Cube (1)

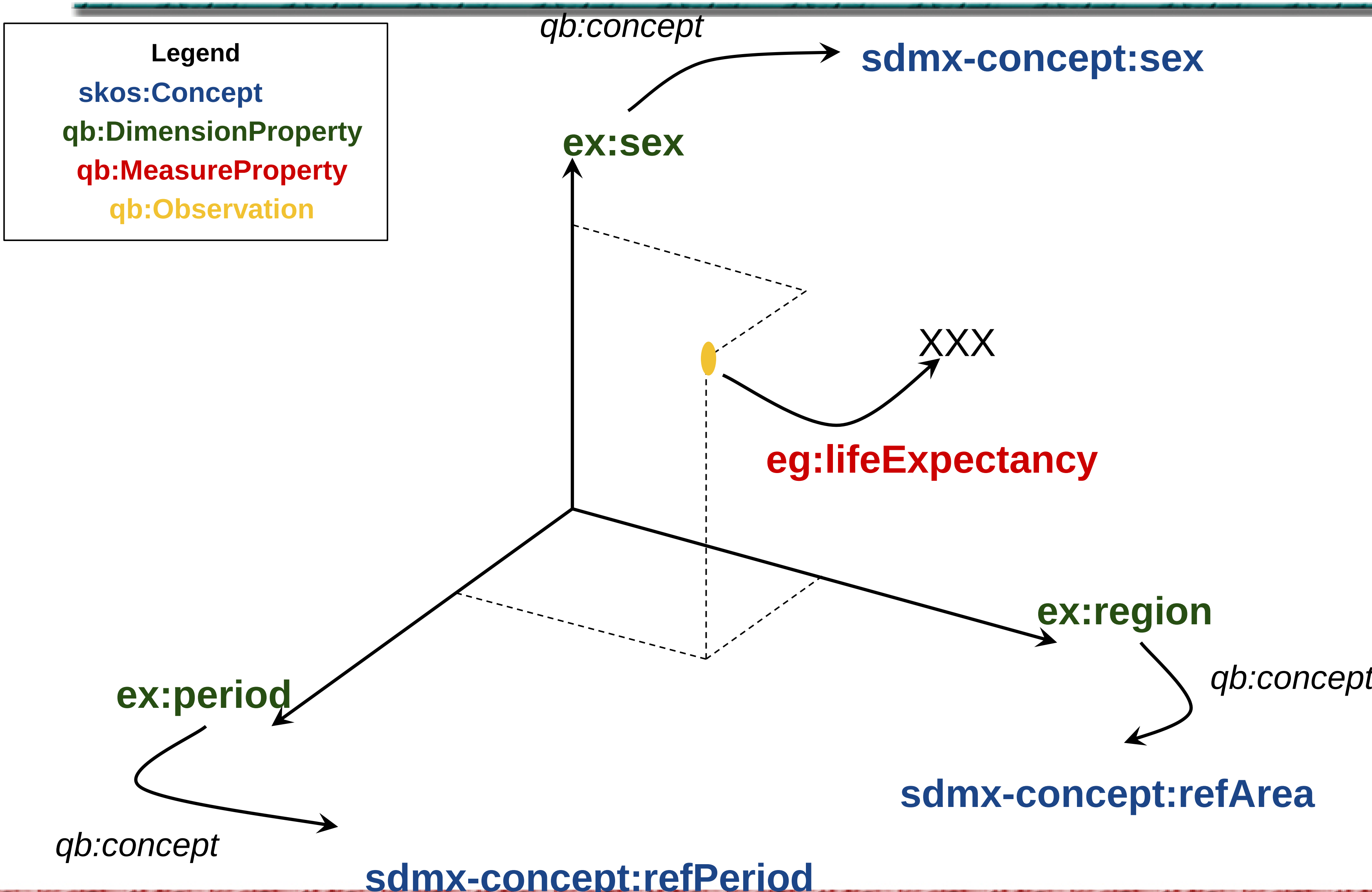
Data Cube is an *RDF vocabulary* (being developed by the W3C Government Linked Data Working Group) for publishing **multidimensional data** (e.g. statistics) on the web of data.

A multi-dimensional data sets comprises a collection of **measurements** made at some point along a group of **dimensions**.

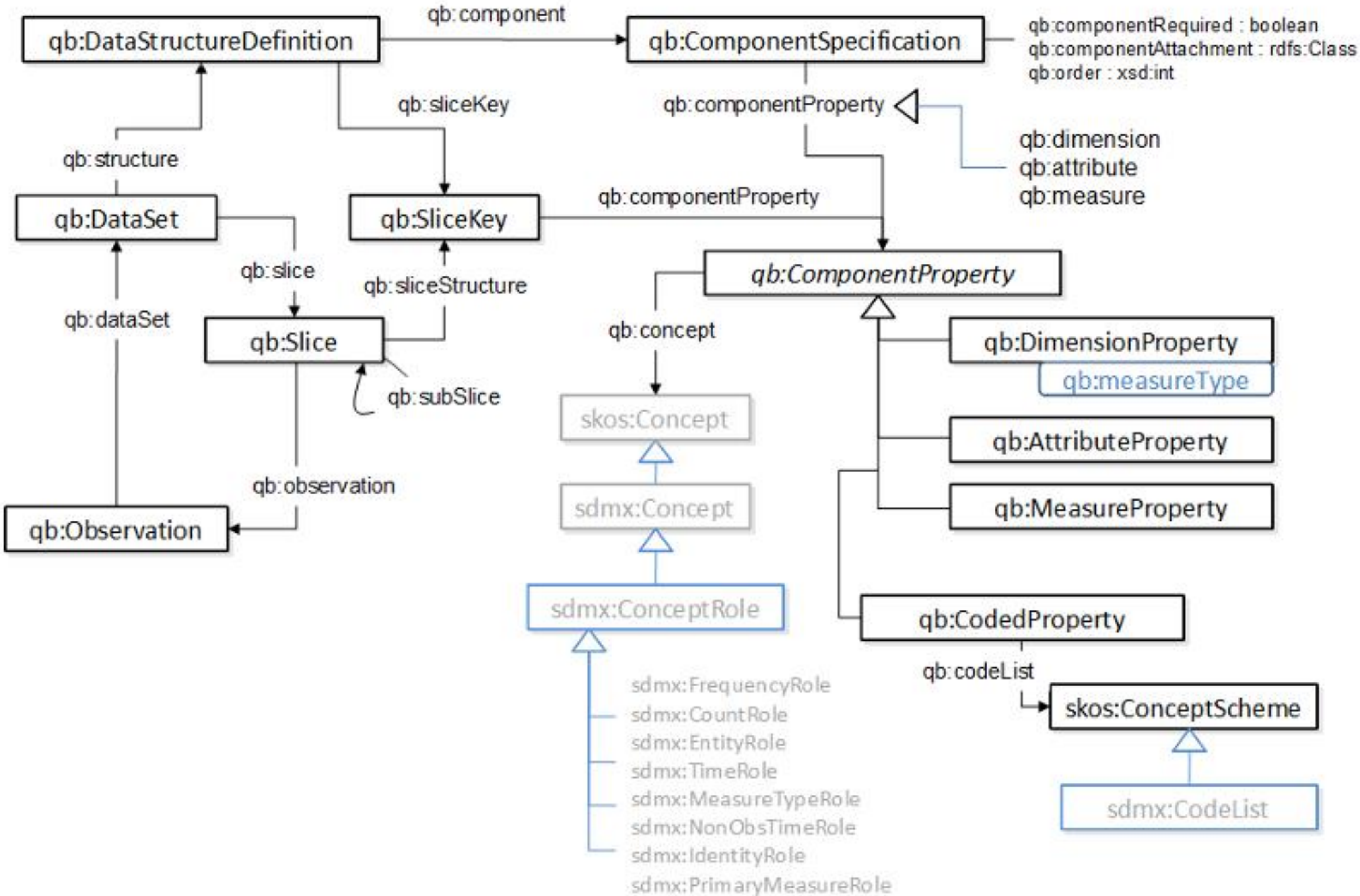
The measures meta-data (such as unit, status, etc.) are expressed by **attributes**.

Dimensions, *attributes* and *measure* are collectively called **components**. Each component may be optionally linked to the concept it expresses. Those concepts have to be SKOS concepts.

SKOS in action - Data Cube (2)



SKOS in action - Data Cube (3)



References (1)

1. Antoine Isaac, and Ed Summers. SKOS Simple Knowledge Organization System Primer. August 2009.
2. Alistair Miles, and Sean Bechhofer. SKOS Simple Knowledge Organization System: Reference. August 2009.
3. Alistair Miles, and Sean Bechhofer. SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL) Namespace Document - HTML Variant. August 2009
4. Graham Klyne, and Jeremy J. Carroll. Resource Description Framework RDF: Concepts and Abstract Syntax. February 2004.
5. Patrick Hayes. RDF Semantics. February 2004.
6. Frank Manola, and Eric Miller. RDF Primer. February 2004.
7. Dan BrickleyBrickley, and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. February 2004.
8. Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language: Primer. October 2009.

References (2)

9. Eric Prud'hommeaux, and Andy Seaborne. SPARQL Query Language for RDF. January 2008

10. Jeni Tennison. Richard Cyganiak, and Dave Reynolds Eds. The RDF Data Cube Vocabulary. April 2012

11. Caterina Caracciolo, Armando Stellato, Ahsan Morshed, Gudrun Johannsen, Sachit Rajbhandari, Yves Jacques and Johannes Keizer. Thesaurus Maintenance, Alignment and Publication as Linked Data - The AGROVOC Use Case, (to appear in) International Journal of Metadata, Semantics and Ontologies, 2012

12. Gail Hodge. Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files. Council on Library and Information Resources, April 2000

13. Nancy Ide, Jean Véronis, and Aix-en-provence Cedex. Machine Readable Dictionaries: What Have We Learned, Where Do We Go?. 1994.

14. Juan-Antonio Pastor-Sanchez, Francisco Javier Martínez Mendez, and José Vicente Rodríguez-Muñoz. Advantages of thesaurus representation using the Simple Knowledge Organization System (SKOS) compared with proposed alternatives.

15. Tim Berners-Lee. Linked Data. July 2007.