Elaboration of controlled vocabularies using SKOS

Juan Antonio Pastor Sánchez
pastor@um.es
University of Murcia (Spain)

DC-2015 São Paulo, Brazil,
September 4th
<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 – 10:00</td>
<td>- The role of SKOS in the Semantic Web and Linked Open Data.</td>
</tr>
<tr>
<td></td>
<td>- SKOS: features and elements</td>
</tr>
<tr>
<td>10:30 – 12:00</td>
<td>- The SKOS-XL extension</td>
</tr>
<tr>
<td></td>
<td>- The SKOS / ISO-THES ontology</td>
</tr>
<tr>
<td></td>
<td>- Quality of SKOS datasets</td>
</tr>
<tr>
<td>14:00 – 15:30</td>
<td>- Tools for working with SKOS</td>
</tr>
<tr>
<td></td>
<td>- Creating a SKOS vocabulary with Protégé</td>
</tr>
<tr>
<td></td>
<td>- Validation of SKOS vocabularies</td>
</tr>
</tbody>
</table>
Three ideas about the Semantic Web

1. **The Semantic Web is not different from the original Web. It's just an approach to publishing, accessing and using structured data.**

2. **The Semantic Web is based on the use of metadata vocabularies and ontologies developed by applying open standards.**

3. **URI references allow to identify both, resources and description elements. This fact conforms the core of the RDF data model used in the Semantic Web.**
Architecture of the Semantic Web

Most apps use only a subset of the stack
Querying allows fine-grained data access
Ontologies allow knowledge semantics representation
A data model for the standardized information exchange is key
Formats (syntax) are necessary, but not too important
The Semantic Web is based on the Web

Content negotiation provides the data in a specific format
URI dereferencing provides access to resources
Linked Data use a small selection of technologies
The RDF data model allows us to make statements about resources (documents, people, physical objects, concepts...) using triples:

\[
<\text{subject}><\text{predicate}><\text{object}>
\]

The object of a statement about a resource can be another resource or data. In the first case the predicate is a “object property”, in the second a “data property”.

The triples define connected graphs. The subjects and objects of the triples are the nodes in the graphs, the predicates are the arcs. Everything (nodes and arcs) are identified with URIs.

DC-2015 São Paulo, Brazil, September 4th.
The RDF data model: example

**Romeo + Juliet**
http://www.wikidata.org/entity/Q463313

cwork:about

**Romeo and Juliet**
http://www.wikidata.org/entity/Q83186

dcterms:title

dcterms:creator

**William Shakespeare**
https://viaf.org/viaf/96994048

"Romeo and Juliet"

**dcterms**: http://purl.org/dc/terms/

**cworks**: http://www.bbc.co.uk/ontologies/creativework/
The RDF data model: serialization

RDF graphs must be serialized to ensure that machines to exchange and process data.

<table>
<thead>
<tr>
<th>RDF/XML</th>
<th>RDF/XML</th>
</tr>
</thead>
</table>
| `<rdf:Description rdf:about="http://www.wikidata.org/entity/Q83186">  
  <dcterms:creator rdf:resource="https://viaf.org/viaf/96994048"/>  
  <dcterms:title>Romeo and Juliet</dcterms:title>  
</rdf:Description>` | Allow the use of XML technologies (XSLT, XPath, XQuery, etc.) |

<table>
<thead>
<tr>
<th>Turtle</th>
<th>Turtle</th>
</tr>
</thead>
</table>
| `<http://www.wikidata.org/entity/Q83186>  
  dcterms:creator <https://viaf.org/viaf/96994048>;  

<table>
<thead>
<tr>
<th>JSON-LD</th>
<th>JSON-LD</th>
</tr>
</thead>
</table>
| `@id:"http://www.wikidata.org/entity/Q83186",  
  "dcterms:creator":https://viaf.org/viaf/96994048",  
  "dcterms:title":{"@language":"en","@value":"Romeo and Juliet"` | Designed for consumption by JavaScript |

<table>
<thead>
<tr>
<th>RDFa</th>
<th>RDFa</th>
</tr>
</thead>
</table>
| `<div resource="http://www.wikidata.org/entity/Q83186">  
  <h1 property="dcterms:title">Romeo and Juliet</h1>  
  <p>Author:<span property="dcterms:creator" resource="https://viaf.org/viaf/96994048">  
  William Shakespeare</span></p>  
</div>` | Allows to insert RDF statements in (X)HTML documents |
The RDF statements, that conform a dataset, can be stored into a text plain file. SPARQL allows to recover sentences that meet certain search conditions. **SPARQL is for RDF like SQL is for the Relational Model.**

```sparql
select ?author
WHERE {
}
```

**Who is the author of “Romeo and Juliet”?**

A SPARQL Endpoint is a RESTful web service that accepts and processes SPARQL sentences.

SPARQL queries retrieve statements stored into a RDF database named “Triplestore”. Results are returned using a format (serialization).
SKOS: Overview

The **Simple Knowledge Organization System** (SKOS) is a **data model** for knowledge organization systems such as thesauri, classifications, subject heading, taxonomies, etc.

SKOS is defined as an OWL ontology and considers a KOS as a **concept scheme comprising a set of concepts** identified unambiguously by URIs.

SKOS concepts can be **labeled** with any number of lexical strings in any given natural language. One of these labels in any given language can be indicated as the **preferred** label for that language, and the others as **alternative** or **hidden** labels.

SKOS concepts can be linked to each other using hierarchical and associative **semantic relations**.

SKOS **concepts can be documented with notes** of various types: scope notes, definitions, editorial notes, etc.

SKOS concepts can be grouped into **collections**, which can be labeled and/or ordered.

SKOS **concepts of different concept schemes can be mapped**. SKOS provides four basic types of mapping link: hierarchical, associative, close equivalent and exact equivalent.
SKOS: Applications

With SKOS, a knowledge organization system can be expressed as **machine-readable data**. It can then be exchanged between computer applications and published in a **machine-readable format** in the Web.

SKOS data are expressed as **RDF triples** and may be encoded using any concrete **RDF syntax**.

SKOS is the **standard** technology to represent a **wide range of knowledge organization systems**.

SKOS can be used in the **indexing** of resources available on the web.

SKOS allows **interconnect** (align) different information systems by **defining equivalences** between knowledge organization systems used in each of them.

The **exploitation of the semantic and mapping relationships**, along with indexing resources with SKOS concepts, are a mechanism of great potential for the **discovery of information**.
SKOS datasets

- STW Thesaurus for Economics
- Thesaurus for the Social Sciences
- DDC Dewey Decimal Classification
- Library of Congress’ vocabularies
- German national Library' subject headings
- French National Library's subject headings
- VIAF person authorities
- Wikipedia categories
- New York Times subjects
- GEMET General Multilingual Environmental Thesaurus
- AGROVOC Agricultural Thesaurus
- Getty Thesaurus of Geographic Names
- Art & Architecture Thesaurus
- UKAT UK Archival Thesaurus
- International Press Telecommunications Council (IPTC) NewsCodes
- Australian education vocabularies
- Proposed international standard nomenclature for fields of science and technology
- UNESCO Thesaurus
- Eurovoc

... and many more

http://datahub.io/dataset?tags=format-skos
Information Discovery with SKOS

Catalog

Subject Headings

Authority

Digital repository

Thesaurus

Wikipedia

Categories

DBPedia

Taxonomy

Web Contents

SKOS DISCOVERY ENGINE

Romeo and Juliet

Search

Title: Romeo and Juliet
Author: William Shakespeare
Description: 131 p. : 21 cm.
Language: English
Notes: p. 131.
Subject: Shakespeare, William, -- 1564-1616. -- Romeo and Juliet.
Author / Creator: Cookson, Linda.
HOLLIS Number: 002290336
Creation Date: 1991
Permalink: http://id.lib.harvard.edu/aleph/002290336/catalog
Locations & Availability

William Shakespeare
Poet

Wikipedia

Romeo + Juliet
1996 film

DC-2015 São Paulo, Brazil, September 4th.
SKOS: Classes

**skos:ConceptScheme**

A SKOS concept scheme can be viewed as an aggregation of one or more SKOS concepts. Concepts can be created and used as stand-alone entities. However, especially in indexing practice, concepts usually come in carefully compiled vocabularies, such as thesauri or classification schemes.

**skos:Concept**

The fundamental element of the SKOS vocabulary is the concept that exist in the mind as abstract entities which are independent of the terms used to label them. A SKOS concept can be viewed as an idea or notion; a unit of thought.

**skos:Collection & skos:OrderedCollection**

SKOS makes it possible to define meaningful groupings or "collections" of concepts. Collections are useful where a group of concepts shares something in common, and it is convenient to group them under a common label, or where some concepts can be placed in a meaningful order.
**SKOS: Labeling properties**

- **skos:prefLabel**
  Assign a preferred lexical label to a resource with an optional language tags. A resource can only have one preferred label per language tag. The preferred label of a concept may also be used to unambiguously represent this concept in a language within a KOS.

- **skos:altLabel**
  Makes it possible to assign alternative lexical labels, beyond the one that is preferred for the concept. Helpful to represent synonyms, near-synonyms, abbreviations and acronyms.

- **skos:hiddenLabel**
  A hidden lexical label for a resource is a character string accessible to applications, but not directly visible for users. Hidden labels may, for example, be used to include misspelled variants of other lexical labels. This way, the user will find a relevant concept if mis-spelled query match against a hidden label.
SKOS: Labeling properties example

```
@prefix ex: <http://www.example.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ex:Brazil rdfs:type skos:Concept ;
    skos:prefLabel "Brazil"@en ;
    skos:prefLabel "Brasil"@pt ;
    skos:altLabel "Federative Republic of Brazil"@en ;
    skos:altLabel "República Federativa do Brasil"@pt .
```
skos:broader & skos:narrower
Allow to assert that one concept is broader or narrower in meaning than another. These properties are each other's inverse. According to the SKOS data model, whenever a concept X is broader than another concept Y, then Y is a narrower concept of X.

skos:broaderTransitive & skos:narrowerTransitive
The properties skos:broader and skos:narrower, are not defined explicitly as transitive. This means that their semantics do not support inferences. For applications that require such semantics SKOS features skos:broaderTransitive and skos:narrowerTransitive.

skos:related
Enables the representation of associative (non-hierarchical) links, such as relationship between one type of event and entities which typically participate in it. This property is symmetric: whenever a concept X is related with another concept Y, then Y is related with X. SKOS does not define this property as transitive.
@prefix ex: <http://www.example.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ex:NAmerica rdfs:type skos:Concept ;
    skos:broader  ex:Americas .
ex:SAmerica rdfs:type skos:Concept ;
    skos:broader  ex:Americas ;
    skos:narrower ex:Brazil .
ex:Americas rdfs:type skos:Concept ;
    skos:narrower ex:NAmerica ;
    skos:narrower ex:SAmerica .
ex:Brazil rdfs:type skos:Concept ;
    skos:broader  ex:SAmerica .
SKOS: Grouping properties

- **skos:inScheme**
  Concepts can be linked to one or more concept schemes using skos:inScheme.

- **skos:member & skos:memberList**
  To indicate the concepts that are part of a collection are used skos:member (collections) and skos:memberList (ordered collections). Collections can be nested with these properties.

- **skos:hasTopConcept**
  The property is used to link a concept scheme to the SKOS concept(s) which are topmost in the hierarchical relations for that scheme.

- **skos:topConceptOf**
  This is the inverse property of skos:hasTopConcept. It indicates when a concept is at the top of the hierarchical structure of a concept scheme.
@prefix ex: <http://www.example.org/> .

[...]

ex:Thesaurus rdfs:type skos:ConceptScheme ;

wikidata:Africa rdfs:type skos:Concept ;
  skos:inScheme ex:Thesaurus .
  skos:topConceptOf ex:Thesaurus

[...]
Prefixes used:

@prefix wikidata : <http://www.example.org/> .

```
@prefix wikidata : <http://www.example.org/> .

ex:ColIntOrg rdfs:type skos:Collection ;
   skos:prefLabel  "Countries by International Organizations"@en ;
   skos:member  ex:ColMercosur, ex:ColEurozone .

ex:ColMercosur rdfs:type skos:Collection ;
   skos:prefLabel  "Mercosur"@en ;

ex:ColEurozone rdfs:type skos:Collection ;
   skos:prefLabel  "Eurozone"@en ;
```
### Documentation properties

- **skos:scopeNote**
  
  Some, possibly partial, information about the intended meaning of a concept, especially as an indication of how the use of a concept is limited in indexing practice.

  ex:microwaveFrequencies skos:scopeNote "Used for frequencies between 1GHz to 300Ghz"@en.

- **skos:definition**
  
  Complete explanation of the intended meaning of a concept.

  ex:documentation skos:definition "the process of storing and retrieving information in all fields of knowledge"@en.

- **skos:example**
  
  Specific examples which refer to a concept.

  ex:organizationsScienceCulture skos:example "academies of science, general museums, world fairs"@en.

- **skos:editorialNote**
  
  Supplies information that is an aid to administrative housekeeping, such as reminders of editorial work still to be done.

  ex:doubleclick skos:editorialNote "Review this term after complete first draft"@en.
  ex:folksonomy skos:editorialNote "Check spelling with Thomas Vander Wal"@en.

- **skos:historyNote**
  
  Describes significant changes to the meaning or the form of a concept.

  ex:childAbuse skos:historyNote "estab. 1975; heading was: Cruelty to children [1952-1975]"@en.

- **skos:changeNote**
  
  Documents fine-grained changes to a concept, for the purposes of administration and maintenance.

  ex:tomato skos:changeNote "Moved from under 'fruits' to under 'vegetables' by Horace Gray"@en.
SKOS: Mapping properties

Every SKOS concept is unambiguously referenced with an URI in any SKOS application.

This is useful for mapping (alignment) links between SKOS concepts in different concept schemes, and their use for information retrieval tools that use several KOSs at the same time.

SKOS allows to specify how the meanings of two concepts from different schemes can be compared.

- **skos:broadMatch & skos:narrowMatch**
  Used to state a hierarchical mapping link between two concepts from different schemes.

- **skos:relatedMatch**
  Allows to state an associative mapping link between two concepts from different schemes.

- **skos:closeMatch**
  Used to link two concepts sufficiently similar that can be used interchangeably in some information retrieval applications.

- **skos:exactMatch**
  Used to link two concepts that can be used interchangeably across a wide range of information retrieval applications. It is a transitive property.
SKOS: Mapping example

@Prefix lem: <http://id.sgcb.mcu.es/Autoridades/> .
@Prefix rameau: <http://stitch.cs.vu.nl/vocabularies/rameau/ark:/> .
@Prefix lcsh: <http://id.loc.gov/authorities/subjects/> .

rameau:12148/cb13318537z rdfs:type skos:Concept ;
   skos:prefLabel "Aide économique"@fr ;

lcsh:sh85040767 rdfs:type skos:Concept ;
   skos:prefLabel "Economic assistance"@en ;
   skos:closeMatch rameau:12148/cb13318537z .

lem:LEM201001188 rdfs:type skos:Concept ;
   skos:prefLabel "Ayuda económica"@es ;
   skos:altLabel "Cooperación Económica"@es ;
   skos:closeMatch rameau:12148/cb13318537z ;
   skos:closeMatch lcsh:sh85040767 .
Some applications require the creation of explicit links between the labels associated with concepts.

The SKOS lexical labeling properties (skos:prefLabel, skos:altLabel, skos:hiddenLabel) cannot be the subject of an RDF statement, because they are just literal strings and not RDF resources. Therefore, a direct relationship cannot be asserted between them.

SKOS eXtension for Labels (SKOS-XL) provides additional support for identifying, describing and linking lexical entities.

- **skosxl:Label**
  This class allows labels to be treated as first-order RDF resources.

- **skosxl:literalForm**
  This property attaches a single RDF literal to instances of skos:Label class.

- **skosxl:prefLabel, skosxl:altLabel, skosxl:hiddenLabel**
  skosxl:Label instances can then be related to concepts using properties.

- **skos:labelRelation**
  This property allows that skosxl:Label instances can be linked together. It is possible to use custom relationships by defining subproperties of skosxl:labelRelation.
SKOS: SKOS-XL example

ex: C04221 rdf:type skos:Concept ;
    skosxl:prefLabel ex:lab04221_1 ;
    skosxl:altLabel ex:lab04221_2 .

ex:lab04221_1 rdf:type skosxl:Label ;
    skosxl:literalForm "USA"@en .
ex:lab04221_2 rdf:type skosxl:Label ;
    skosxl:literalForm "United States of America"@en .

ex:acronym rdfs:subPropertyOf skosxl:labelRelation .
ex:lab04221_1 ex:acronym ex:lab04221_2 .

@Prefix rdf: <http://id.sgcb.mcu.es/Autoridades/> .
@Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@Prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@Prefix skosxl: <http://www.w3.org/2008/05/skos-xl#> .
@Prefix ex: <http://www.example.org/>
ISO 25964 standard

- ISO 25964 defines a data model to represent thesauri and updates the previous ISO-2788/5964.

- Define higher order structures to model microthesauri, arrays of thesaurus and concept groups.

- Allows modeling hierarchical and associative relationships of an extensible and adaptable way to needs that cover a wide variety of use cases.

- Lexical Compositionality for labeling concepts.

- Defines mapping relationships between different elements of knowledge organization systems, including concepts compositionality.
ISO 25964 UML data model
ISO 25964 and SKOS: ISO-THES Ontology

- The ISO-THES ontology reuses SKOS and SKOS-XL, and defines new properties to represent KOS with RDF by applying the ISO-25964 data model.

- SKOS and SKOS-XL are reused to represent concepts, concept schemes, collections, labeling, notation, documentation, semantic relationships and mapping relationships.

- Two new grouping structures are defined: "Thesaurus Array" and "Concept Group", along with those properties that allow linking it with concepts and concept schemes.

- Generic, Partitive and Instantial hierarchical relationships are defined as sub-properties of skos:broader and skos:narrower.

- SKOS-XL is essential, along with other properties, to define the elements that can represent the compound equivalence between preferred terms and non-preferred terms.
Aplication example of the ISO-THES Ontology

One classic scenario for the application of the ISO-THES Ontology is the modeling of micro-thesauri (i.e.: Eurovoc, UNESCO Thesaurus):

UNESCO THESAURUS → SCIENCE (field) → BIOLOGY (micro-thesaurus)

@prefix ex: <http://www.example.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix iso-thes: <http://purl.org/iso25964/skos-thes#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ex:unescoThes rdfs:type skos:ConceptScheme .

ex:scienceField rdfs:type skosthes:ConceptGroup ;
  skosthes: microThesaurusOf ex:unescoThes ;

ex:biologyMicroThes rdfs:type iso-thes:ConceptGroup ;
  iso-thes: superGroup ex:scienceField .
Data on the Web Challenges Best Practices (1/2)

- **Metadata: Metadata for humans & machines**

- **Data License: How do I permit & restrict access?**
  BP5: Provide license information.

- **Provenance & quality: How can I add trust?**
  BP6: Provide provenance information. BP7: Provide data quality information.

- **Data Versioning: Tracking versions & series**
  BP8: Provide versioning information. BP9: Provide version history.

- **Data Identification: Identifying datasets & distributions**
  BP10: Use persistent URIs as identifiers. BP11: Assign URIs to dataset versions and series.

- **Data Formats: What data formats should I use?**
Data on the Web Challenges Best Practices (2/2)

Data Vocabularies: How to achieve semantic interoperability

Sensitive Data: Privacy & security

Data Access: Access options
BP23: Provide bulk download. BP24: Follow REST principles when designing APIs. BP25: Provide real time access. BP26: Provide up to date data. BP27: Maintain separate versions for data APIs.

Data Preservation: How can data be be archived?

Feedback: How can you engage users?
BP31: Gather feedback from data consumers. BP32: Provide information about feedback.

Data Enrichment: Adding value to data
BP33: Enrich data through generated metadata.
### Quality of SKOS Datasets: Overview

<table>
<thead>
<tr>
<th></th>
<th>Class &amp; property definition</th>
<th>Integrity conditions</th>
<th>Quality issues</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class definitions and properties that must be respected into the dataset to fulfill the SKOS recommendation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrity conditions to be met for a minimum of consistency of the dataset according to the SKOS recommendation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues to be avoided to ensure an optimum quality of SKOS controlled vocabularies.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantitative evaluation of structure and elements of the SKOS dataset.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quality of SKOS Datasets
Definitions and conditions of integrity of the SKOS recommendation

- **Range of skos:inScheme**
- **Domain and range of skos:hasTopConcept**
- skos:ConceptScheme, skos:Concept, skos:Collection and skosxl:Label are disjoint from each other
- **Range of**
  - skos:prefLabel, skos:altLabel y skos:hiddenLabel
  - skos:prefLabel, skos:altLabel and skos:hiddenLabel disyuntas are pairwise disjoint in order to avoid clash between labels of the same language tag.
- A resource has no more than one value of skos:prefLabel per language tag.
- **Domain and range of skos:semanticRelation**
  - Two concepts can not relate to each other by skos:related and skos:broader/narrower simultaneously

- **Domain and range of skos:member**
- **Domain and range of skos:memberList**
  - skos:memberList does not have more than one value
  - Two concepts can not relate to each other by more than one mapping relationship.
- **Domain and range of skosxl:literalForm**
  - skosxl:Label have exactly one skosxl:literalForm
  - Range of skosxl:prefLabel, skosxl:altLabel y skosxl:hiddenLabel
  - skosxl:prefLabel, skosxl:altLabel and skosxl:hiddenLabel are pairwise disjoint in order to avoid clash between labels of the same language tag.

---

- Class & property definition
- Integrity condition
Quality of SKOS Datasets

Quality issues

Q1. Language tags, invalid or missing
Q2. All concepts are not labeled in every language of the vocabulary
Q3. Undocumented Concepts
Q4. Partially documented concepts
Q5. Concepts, Concept Schemes or Collection with missing labels
Q6. Existence of extra white space on labels
Q7. Isolated concepts without semantic relationships
Q8. Isolated cluster concepts
Q9. Cyclical hierarchical relationships
Q10. Useless associative relationships
Q11. Exclusive use of transitive hierarchical relationships
Q12. Unidirectional semantic relationships
Q13. Omitted Concept Scheme
Q14. Omitted Top Concepts
Q15. Top Concepts having broader concepts
Q16. Unmarked Top Concepts
Q17. Unidirectional Related Concepts
Q18. An associative relationship between two concepts, that are part of the same cluster of concepts hierarchically linked, should not be defined
Q19. Missing In-links
Q20. Missing Out-links
Q21. Broken links
Q22. Invalid URIs
Q23. HTTP URI Scheme Violation (not derreferenciables URIs)
Q24. Invalid/Undefined SKOS resources
Q25. Duplicated notations in a Concept Scheme
Quality of SKOS Datasets

Metrics

- **M1** Number of concepts
- **M2** Number of collections
- **M3** Number of Concept Scheme
- **M4** Number of preferred labels per language
- **M5** Number of alternative labels per language
- **M6** Number of hidden labels per language
- **M7** Equivalence rate: ratio between the number of alternative and preferred labels per language.
- **M8** Connection rate: ratio between the number of semantic related concepts and the total number of concepts.
- **M9** Reciprocity rate: ratio between the number of inverse semantic relationships and the total number of the semantic relationships.
- **M10** Enrichment rate: ratio between the sum of the hierarchical and associative relationships and the total number of concepts.
- **M11** Precoordination rate: ratio between the number of keywords of the preferred labels and the total number of concepts.
- **M12** Documentation rate: ratio between the number of significative documentation properties and the total number of concepts.
- **M13** Flexibility rate: ratio between the number of unique keywords of the preferred labels and the number of that keywords presents on the non-composed preferred and alternative labels.
- **M14** Number of hierarchical levels.
- **M15** Size (total number of concepts) of the hierarchical clusters per top concept.
- **M16** Ambiguity rate: ratio between the number of concepts without hierarchical and documentation notes and the total number of concepts.
- **M17** Mapping rate: ratio between the number of concepts with any mapping relationship and the total number of concepts.
**Quality of SKOS Datasets**

**Evaluation model structure**

| Groups of analysis for different aspects of modeling and managing of the vocabulary. |
| Type of vocabulary resources | Labeling and documentation | Macro-structure | Semantic relationships | Mapping and Linked Open Data |

**Level 1**
- Normative aspects and quality criteria for a correct modeling of the vocabulary.

**Level 2**
- Clear and efficient access to vocabulary elements by people and machines.

**Level 3**
- Added value to the representation of SKOS vocabulary.

**Metrics**
- Integration of the metrics according to their relevance with the groups of analysis.
Quality of SKOS Datasets
Evaluation model structure

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Type of vocabulary resources</th>
<th>Labeling and documentation</th>
<th>Macro-structure</th>
<th>Semantic relationships</th>
<th>Mapping and Linked Open Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S9  S37 S48 Q24</td>
<td>S12 S13 S14 S50 S51 S52 S54 S58 S60 S61 Q1 Q24 Q25</td>
<td>S4 S5 S6 S31 S32 S33 S34 S35 Q13 Q14 Q24</td>
<td>S19 S20 S27 Q7 Q9 Q15 Q18 Q24</td>
<td>S46 Q22 Q24</td>
</tr>
<tr>
<td>Level 2</td>
<td></td>
<td>Q2 Q5</td>
<td>Q8 Q14 Q16</td>
<td>Q10 Q12 Q21</td>
<td>Q21</td>
</tr>
<tr>
<td>Level 3</td>
<td></td>
<td>Q3 Q4 Q6</td>
<td>Q11</td>
<td>Q19 Q20 Q23</td>
<td></td>
</tr>
<tr>
<td>Metrics</td>
<td>M1  M2 M3</td>
<td>M4  M5 M6 M7 M11 M12 M13</td>
<td>M8 M15</td>
<td>M8 M9 M10 M14 M15 M16</td>
<td>M17</td>
</tr>
</tbody>
</table>

- Integrity condition
- Class & property definition
- Quality issues
- Metrics

DC-2015 São Paulo, Brazil, September 4th.
SKOS management tools: iQvoc

Open source web application for SKOS vocabulary management. It's a tool that combines easy-to-use human interfaces with Semantic Web interoperability.

- Import of existing vocabularies from a SKOS representation.
- Multilingual display and navigation in any Web browser.
- Editorial features for registered users
- Publishing the vocabulary in the Semantic Web as linked data.
- Easily customized.

http://iqvoc.net/
SKOS management tools: IQvoc

Model building Concept

Assigned collections

Broader terms

Achievement hobbies

Narrower terms

Model rocketry
Radio-controlled modeling
Scale modeling

Definitions
SKOS management tools: PoolParty

Commercial web application designed for the Management of multi-lingual (corporate) thesauri & taxonomies with Semantic Web standards and thesaurus quality checker.

- Web-based AJAX user interface, Drag & drop, Auto-Complete
- Document analysis: phrase extraction
- Publish thesauri as linked data & Enrich concepts by using linked data
- Advanced reporting functionality, Import and validation of thesauri and CSV files
- RDF Triple Store index engine and a phrase-extraction component
- Wiki style collaborative editing of thesauri & quality checker
- Java-API, HTTP web services, SPARQL endpoint, Linked Data

https://www.poolparty.biz/
SKOS management tools: PoolParty
SKOS management tools: TemaTres

It's an open source vocabulary server, web application to manage and exploit vocabularies, thesauri, taxonomies and formal representations of knowledge.

Ensures consistency and integrity of data and relationships between terms.

Enables vocabularies to be represented in a wide range of metadata standards.

Service oriented Architecture to share and re-use controlled vocabularies and SPARLQ Endpoint.

Web services capabilities and specialized API focused in terminological issues and concept representations in multiple formats: XML, RDF/XML, JSON.

Multiple publishing options and integration with WordPress.

http://www.vocabularyserver.com/
### Unesco Thesaurus

**COSMOLOGY**

<table>
<thead>
<tr>
<th>Term</th>
<th>Notes (1)</th>
<th>Metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSMOLOGY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Broader Terms**
- BT ASTRONOMY

**Related terms**
- RT METAPHYSICS
- RT SPACE SCIENCES
- RT UNIVERSE

**Español** COSMOLOGÍA

**Français** COSMOLOGIE
Protégé Desktop as SKOS editor

Protégé is a free, open source ontology editor and a knowledge acquisition system. It provides a graphic user interface to define ontologies.

Therefore, Protégé is not a specialized SKOS editor, but a generic ontology editor. However, it allows the creation of SKOS vocabularies with a few basic functionality.

It is a good tool for basic/initial design of SKOS vocabularies in a local environment.

Labeling and documentation properties of SKOS are “Annotation properties”.

The semantic relations are “Object properties”.

Notations and SKOS-XL literals are “Data properties”.

The Concept Schemes, Concepts, Collections and SKOS-XL Labels are created as instances (individuals) of their respective classes.
Downloading and installing Protégé

 downloading and installing Protégé

Protoé Desktop is a feature rich ontology editing environment with full support for the OWL 2 Web Ontology Language, and direct in-memory connections to description logic reasoners like HermitT and Pellet.

Protoé Desktop supports creation and editing of one or more ontologies in a single workspace via a completely customizable user interface. Visualization tools allow for interactive navigation of ontology relationships. Advanced explanation support aids in tracking down inconsistencies. Refactor operations available including ontology merging, moving axioms between ontologies, rename of multiple entities, and more.

- W3C standards compliant
- Customizable user interface
- Visualization support
- Ontology refactoring support
- Direct interface to reasoners
- Highly pluggable architecture
- Cross compatible with WebProtoé

Download for Linux
Protoé Desktop 5.0 beta

Download platform Independent version (requires a Java Runtime Environment)

http://protege.stanford.edu/products.php#desktop-protege

Include a Java Runtime Environment

Compact installation but JRE must be present in the OS
Example: Earth and Space Science Thesaurus

Earth and Space Science Thesaurus
  Atmospheric sciences
  Climatology
  Geochemistry
  Geodesy
  Geography
  Geology
    Engineering geology
    Environmental geology
    Geohydrology
  Geophysics
  Hydrology
    Erosion by water
    Evaporation
    Hydrobiology
    Hydrography
  Meteorology
  Oceanography
  Soil Science
  Space Science
Example: Earth and Space Science Thesaurus

- Atmospheric sciences
- Climatology
- Geochemistry
- Geodesy
- Geography
- Geology
  - Engineering geology
  - Environmental geology
  - Geohydrology
- Geophysics
- Hydrology
  - Erosion by water
  - Evaporation
  - Hydrobiology
  - Hydrography
- Meteorology
- Oceanography
- Soil Science
- Space Science

The thesaurus are represented with skos:ConceptScheme.

Descriptors are modeled as skos:Concept and interrelated through hierarchical semantic relations with the properties skos:broadernarrower.
Example: Earth and Space Science Thesaurus

Earth and Space Science Thesaurus / Tesauro de Ciencias de la Tierra y del Espacio
Atmospheric sciences / Ciencias de la atmósfera
Climatology / Climatología
Geochemistry / Geoquímica
Geodesy / Geodesia
Geography / Geografía
Geology / Geología
  Engineering geology / Geología aplicada a la ingeniería
  Environmental geology / Geología ambiental
  Geohydrology / Hidrogeología
Geophysics / Geofísica
Hydrology / Hidrología
  Erosion by water / Erosión por agua
  Evaporation / Evaporación
  Hydrobiologist / Hidrobiología
  Hydrography / Hidrografía
Meteorology / Meteorología
Oceanography / Oceanografía
Soil Science / Ciencias del Suelo
Science space / Ciencias del Espacio

Descriptors are represented with the property skos:prefLabel. A concept have one preferred label per language tag.

English / Spanish

English / Spanish
Example: Earth and Space Science Thesaurus

Earth and Space Science Thesaurus / Tesauro de Ciencias de la Tierra y del Espacio
- Atmospheric sciences / Ciencias de la atmósfera
- Climatology / Climatología
- Geochemistry / Geoquímica
- Geodesy / Geodesia [Alternative: Earth measurement / Medición de la Tierra]
- Geography / Geografía
- Geology / Geología
  - Engineering geology / Geología aplicada a la ingeniería
  - Environmental geology / Geología ambiental
  - Geohydrology / Hidrogeología
- Geophysics / Geofísica
- Hydrology / Hidrología
  - Erosion by water / Erosión por agua
  - Evaporation / Evaporación
  - Hydrobiology / Hidrobiología
- Hydrography / Hidrografía [Alternative: Hydrology / Hidrología]
- Meteorology / Meteorología
- Oceanography / Oceanografía [Alternative: Marine science, Oceanology / Ciencias del Mar]
- Soil Science / Ciencias del Suelo [Alternative: Edaphology / Edafología]
- Space Science / Ciencias del Espacio

Non-Descriptors are modeled using the skos:altLabel property. A concept can have more than one alternative label per language tag. The number of alternative labels per language tag in the concept can be different.
Example: Earth and Space Science Thesaurus

Earth and Space Science Thesaurus / Tesauro de Ciencias de la Tierra y del Espacio
Atmospheric sciences / Ciencias de la atmósfera [Related: Climatology, Meteorology]
Climatology / Climatología [Related: Atmospheric sciences, Meteorology]
Geochemistry / Geoquímica
Geodesy / Geodesia [Alternative: Earth measurement / Medición de la Tierra]
Geography / Geografía
Geology / Geología
  Engineering geology / Geología aplicada a la ingeniería
  Environmental geology / Geología ambiental
  Geohydrology / Hidrogeología [Related: Hidrology]
Geophysics / Geofísica
Hydrology / Hidrología [Related: Geohydrology]
  Erosion by water / Erosión por agua
  Evaporation / Evaporación
  Hydrobiology / Hidrobiología
  Hydrography / Hidrografía [Alternative: Hydrology / Hidrología]
Meteorology / Meteorología [Related: Atmospheric sciences, Climatology]
Oceanography / Oceanografía [Alternative: Marine science, Oceanology / Ciencias del Mar]
Soil Science [Alternative: Edaphology / Edafología]
Space Science / Ciencias del Espacio

For Related concepts the property skos:related must be used. This property is symmetric, therefore the relationship must be defined in both directions:

<A> skos:related <B>
<B> skos:related <A>

http://skos.um.es/workshop/workshop-thesaurus.txt

Alternative labels
Related concepts
Creating the thesaurus based in SKOS with Protégé

1. Set the ontology URI.
2. Import the SKOS Ontology.
3. Create the Concept Scheme to represent the thesaurus using the skos:ConceptScheme class and their annotation properties (dcterms:title and skos:prefLabel).
4. Create the concepts with the skos:Concept class and assign the corresponding SKOS labelling properties (preferred, alternative and hidden) in every language tag.
5. Set the top concepts of the Concept Scheme with the skos:hasTopConcept property.
6. Set the hierarchical structure of concepts from top to down with skos:broader / skos:narrower.
7. Set the associative relationships using the skos:related property.
Set the Ontology URI

In this example the concepts will have URIs of the type:
http://www.example.org/concept_04332

The Ontology URI allows to define de URI base of the KOS
Import the SKOS Ontology

The ISO-THES ontology import indirectly both, SKOS and SKOS-XL

The ISO-THES ontology import indirectly both, SKOS and SKOS-XL

Please specify the URL that points to the file that contains the ontology (this should be the physical URL, rather than the ontology URI)

URI: http://purl.org/iso25964/skos-thes#

Ontology IRI: <http://purl.org/iso25964/skos-thes>  
Location: http://purl.org/iso25964/skos-thes#

Indirect Imports:

<http://www.w3.org/2008/05/skos-xl>  
skosxl
Ontology IRI: <http://www.w3.org/2008/05/skos-xl>  
Location: http://www.w3.org/2008/05/skos-xl

<http://www.w3.org/2004/02/skos/core>  
skos
Ontology IRI: <http://www.w3.org/2004/02/skos/core>  
Location: http://www.w3.org/2004/02/skos/core
Import the SKOS Ontology

The elements of the imported ontologies can be checked (even edited) under the “Entities” tab.

By default, the “Entities” tab have a panel with the class hierarchy.

The panel below contains all the properties (object, data and annotation) among available datatypes and the individuals by type (class).
Prefixes

The ontology prefixes will simplify the management of the vocabulary.

Now is the perfect time to save the ontology: File → Save as. For this workshop we use Turtle for the file format.
New entities preferences

The preferences dialog box is available from File → Preferences.

This option defines the type of cool URI used by the dataset.

The ontology URI is used as prefix for the URIs of the new entities.

The ontology URI is used as prefix for the URIs of the new entities.

With this option it is possible to label the entity at the time of its creation.

This area allows to define the pattern for the ID of the entity.
Creating the elements of the Thesaurus

The elements of the thesaurus are created as entities (individuals) of the ontology.

Different classes of SKOS, SKOS-XL and ISO-THES are available in the “Class hierarchy” panel.

The panel “Instances” shows the elements by type (class).

The “Annotations” panel allows to manage the labelling properties.

The “Property assertions” panel allows to define new relationships (“Object property assertions”).
Creating the Concept Scheme

The skos:ConceptScheme class must be selected.

These buttons "Add" or "Delete" instances.

After press "Add" button the dialog box for creating new instances will appear.

With the preferences for new entities of the previous slide a skos:prefLabel property will be created using the name entered in this field.
Labelling the Concept Scheme

The text entered in the Name field is used as preferred label. Other preferred, alternative labels can be created with the “Add” button.

In addition, the concept schemes are usually labelled with the Dublin Core term “title”.

Both, label displayed in the Instances panel and order of Annotations panel can be modified in View → Custom rendering.

The “Add” button creates new annotations properties.
If needed it's possible change the entity URI with Refactor → Rename Entity
Creating and labelling the Concepts

The skos:Concept class must be selected

The labelling properties can be created from the Annotations panel
Defining the Thesaurus Top Concepts

The `skos:ConceptScheme` class must be selected.

The instance of the Thesaurus Concept Scheme must be selected.

The `skos:hasTopConcept` link a concept scheme with one or more concepts. Therefore, it's a object property.
Selecting the Thesaurus Top Concepts

The skos:hasTopConcept object property must be selected.

Select all the Thesaurus Top Concepts.

One skos:hasTopConcept property for the Concept Scheme are defined for every selected concept.
Hierarchical relationships

The skos:Concept class must be selected.

The instance of the Concept must be selected.

The skos:narrower link a concept with another/s concept/s. Therefore, it's a object property.
Selecting the narrower concepts

The skos:narrower object property must be selected. Note that skos:narrower is a sub-property of skos:narrowerTransitive.

Select all the narrower concepts for “Geology”.

One skos:narrower property for the Concept “Geology” are defined for every selected concept.
Associative relationships

The instance of the source concept for the associative relationship must be selected.

The skos:related object property must be selected.

Select the target concept for the associative relationship.

The skos:related property for the Concept "Geohydrology" is defined.
Since Protégé is not a specialized editor, there are certain tasks that are not performed automatically.

Protégé does not define the inverse relationships automatically: broader ↔ narrower, related ↔ related

Protégé does not check inconsistencies in the labeling or the structure of semantic relations.

Protégé does not define the skos:inScheme property for the concepts with the concept scheme.

http://skos.um.es/workshop/ws.ttl

[...]

```ttl
### http://www.example.org/concept_00022
ex:concept_00022 rdf:type owl:NamedIndividual ,
    skos:Concept ;
    skos:prefLabel "Geohydrology"@en , "Hidrogeología"@es ;
    skos:related ex:concept_00024 .

### http://www.example.org/concept_00023
ex:concept_00023 rdf:type owl:NamedIndividual ,
    skos:Concept ;
    skos:prefLabel "Geophysics"@en , "Geofísica"@es .

### http://www.example.org/concept_00024
ex:concept_00024 rdf:type owl:NamedIndividual ,
    skos:Concept ;
    skos:narrower ex:concept_00025 ,
    ex:concept_00026 ,
    ex:concept_00027 ,
    ex:concept_00028 .

### skos:related ex:concept_00022 should be here

### http://www.example.org/concept_00025
ex:concept_00025 rdf:type owl:NamedIndividual ,
    skos:Concept ;
    skos:prefLabel "Erosion by water"@en , "Erosión por agua"@es .

### skos:broader ex:concept_00024 should be here
```

DC-2015 São Paulo, Brazil, September 4th.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omitted or Invalid Language Tags</td>
<td>only when the language of untagged literals is known and different languages have not been mixed.</td>
</tr>
<tr>
<td>Missing Labels in concept schemes</td>
<td></td>
</tr>
<tr>
<td>Inconsistent Preferred Labels</td>
<td>retain the shortest label when a concept have severral preferred label and convert the rest to alternative labels.</td>
</tr>
<tr>
<td>Disjoint Labels Violation</td>
<td>when several labeling properties use the same label the less important property are removed (hiddenlabel &lt; altLabel &lt; preferredLabel).</td>
</tr>
<tr>
<td>Extra Whitespace in Labels</td>
<td></td>
</tr>
<tr>
<td>Detect (and optionally correct) Cyclic Hierarchical relations</td>
<td></td>
</tr>
<tr>
<td>Detect and correct the omitted top concepts</td>
<td>detecting the concepts having no broader relationships and the missing links from the concepts to the concept scheme (skos:inScheme).</td>
</tr>
<tr>
<td>Enrich the SKOS vocabulary with bidirectional relationships</td>
<td>when unidirectionally related concepts are detected.</td>
</tr>
<tr>
<td>Remove the relation clashes</td>
<td></td>
</tr>
<tr>
<td>Remove the disjoint classes violations</td>
<td></td>
</tr>
</tbody>
</table>
Skosify quality improvement tool (2/3)

Select a SKOS file to upload (maximum size: 20480 KB)
Input file: dcmi-2015-skos-workshop.ttl

Validity options
- Keep skos:related relationships within the same hierarchy
- Break any cycles in the skos:broader hierarchy

Output options
- Include skos:narrower relations in output
- Include transitive hierarchical relations in output

Status: finished
Phase 1 Parsing input files
Phase 2 Performing inferences
Phase 3 Setting up namespaces
Phase 4 Transforming concepts, literals and relations
Phase 5 Performing SKOS enrichments
Phase 6 Cleaning up
Phase 7 Setting up concept schemes and top concepts
Phase 8 Checking concept hierarchy
Phase 9 Checking labels
Phase 10 Writing output

Results
- Processed vocabulary
- Full report

http://demo.seco.tkk.fi/skosify/skosify
Create the inverse relationships for skos:narrower and skos:related

Create the link between the concepts and the concept scheme with skos:inScheme

Define the top concepts of the concept scheme with skos:topConceptOf property

http://skos.um.es/workshop/output.ttl
Notes about the future

- SKOS is a strongly consolidated standard, simple and widely used in numerous datasets.

- Discontinuity in projects for SKOS management and application tools, especially open source and developed with PHP (which is the major development platform for web applications).

- SKOS is not perfect. It needs to be revised to incorporate certain features that improve the linked data consumption (inverse properties of skos:member, skos:inScheme and iso-thes:microThesaurusOf) and the modeling of grouping structures adapted to more case of use (i.e. define certain concepts as points of access to collections).

- KOS interconnection by aligning vocabularies, using mapping relationships, is a territory to explore in depth and can show the true potential of SKOS.
The End

Thanks for your attention...

... and now...

- Questions?
- Suggestions?
- Criticisms?
- Congratulations?
Just a few references

**SKOS Simple Knowledge Organization System Primer**
W3C Working Group Note 18 August 2009
http://www.w3.org/TR/skos-primer/

**SKOS Simple Knowledge Organization System Reference**
W3C Recommendation 18 August 2009
http://www.w3.org/TR/skos-reference/

**ISO 25964 SKOS extension (iso-thes)**
http://pub.tenforce.com/schemas/iso25964/skos-thes/

**Correspondence between ISO 25964 and SKOS/SKOS-XL Models**
http://www.niso.org/schemas/iso25964/correspondencesSKOS/

Mader, C. & Suominen, O. (2014). **Assessing and Improving the Quality of SKOS Vocabularies**
Journal on Data Semantics, 3(1), 47-73.
eprints.cs.univie.ac.at/3707/1/skosquality.pdf

**Data on the Web Best Practices**
W3C Working Draft 25 June 2015
http://www.w3.org/TR/dwbp/