Exploring existing ontologies

FinanceIT@UNSW

Madhushi Bandara and Fethi Rabhi

Never Stand Still

Faculty of Engineering

Computer Science and Engineering

Creating and Publishing Ontologies

Ontologies can be created

- For a very specific purpose
- For a large community of users
- For a particular domain of application
- For a company

Ontologies

- Can redefine concepts with a different meaning
- Can use concepts from other ontologies
- Can be combined

Ontologies

- There are many ontologies available on the Web:
  - General information
  - Health
  - Finance

- We will present and explore a sample of published ontologies:
  - SKOS
  - RDF Cube
  - FIBO

Other important ontologies

- Dublin Core Terms (DCMI)
  - Defines terms: which help determine origin of documents

Content	**Intellectual Property**	**Instantiation**
Coverage	Contributor	Date
Description	Creator	Format
Type	Publisher	Identifier
Relation	Rights	Language
Source		
Subject		
Title		

- FOAF (Friend of a Friend): people and their connections
- DBPEDIA: known people and places

SKOS
(Simple Knowledge Organization System)

Description

- SKOS (the Simple Knowledge Organization System) is a W3C Recommendation that provides a means for representing knowledge organization systems (including controlled vocabularies, thesauri, taxonomies, and folksonomies) in a distributed and linkable way
- SKOS was not designed to replace any thesaurus standard but in fact to augment them by bringing the distributed nature of the SemanticWeb to thesauri and controlled vocabularies

Basic Principles

SKOS consider vocabularies, thesauri, taxonomies as sets of concepts identified with IRIs and grouped into a **concept scheme**

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

Concepts can be linked to each other using hierarchical and associative **semantic relations**

Concepts can be grouped into collections, which can be labeled and/or ordered.

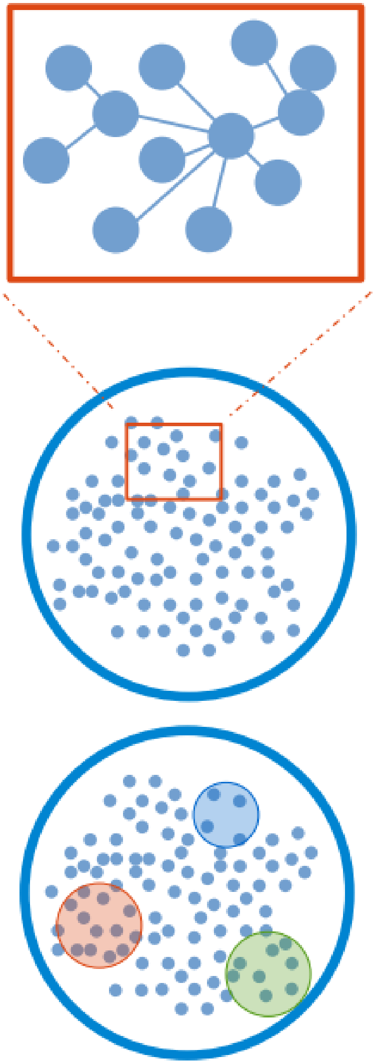
Concepts of different concept schemes can be mapped using four basic types of mapping link: hierarchical, associative, close equivalent and exact equivalent.

SKOS Classes

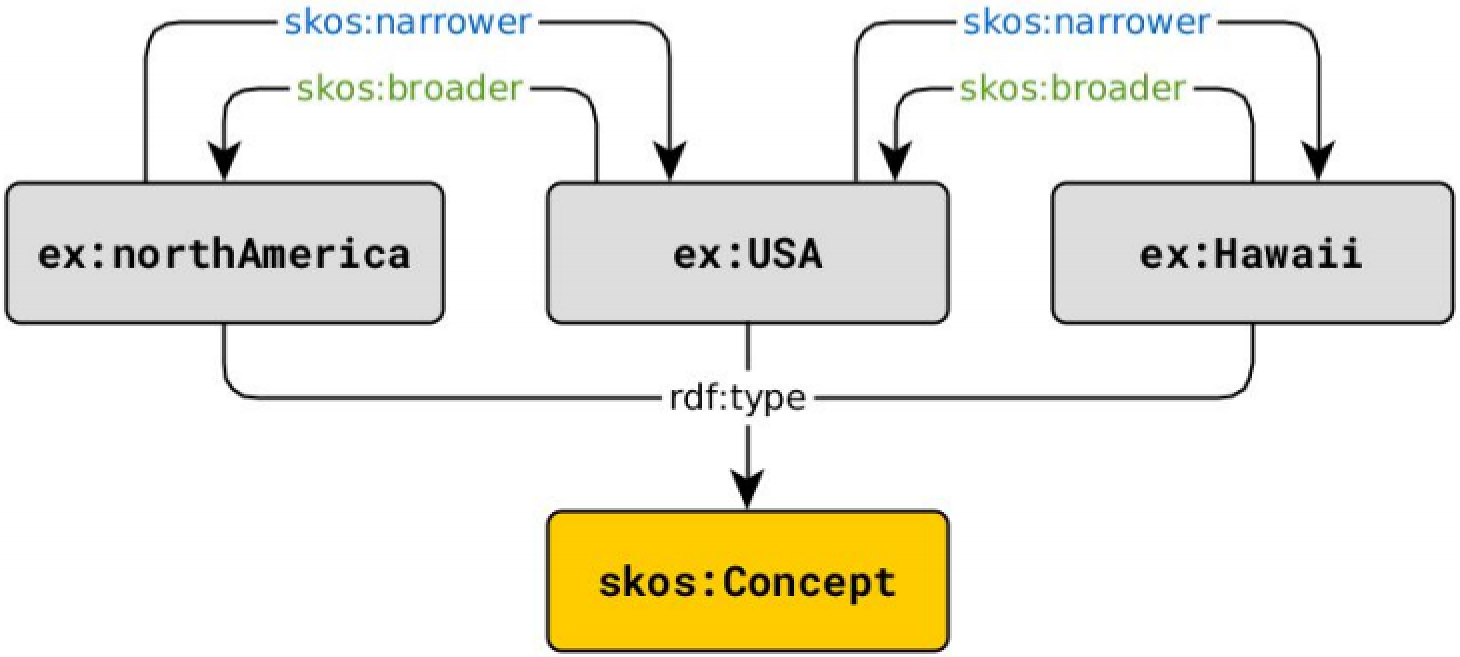
skos:Concept: exists in the mind independently of the terms used to label them. A SKOS concept can be viewed as an idea or notion; a unit of thought.

skos:ConceptScheme: an aggregation of one or more SKOS concepts.

skos:Collection and skos:OrderedCollection SKOS: makes it possible to define meaningful groupings or "collections" of concepts.



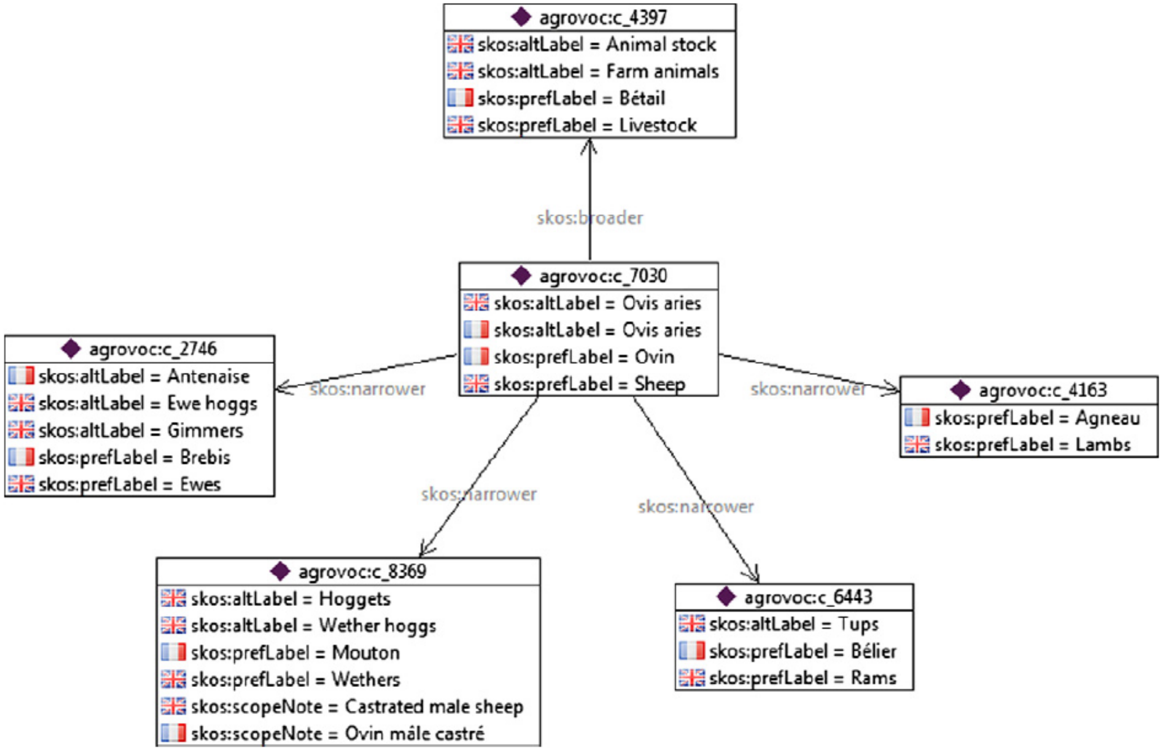
Semantic relations



Example

The United Nations Food and Agriculture Organization

has maintained a thesaurus called AGROVOC for organizing documents about agriculture



agrovoc:c_4397
🇬🇧 skos:altLabel = Animal stock
🇬🇧 skos:altLabel = Farm animals
🇫🇷 skos:prefLabel = Bétail
🇬🇧 skos:prefLabel = Livestock

agrovoc:c_7030
🇬🇧 skos:altLabel = Ovis aries
🇫🇷 skos:altLabel = Ovis aries
🇫🇷 skos:prefLabel = Ovin
🇬🇧 skos:prefLabel = Sheep

agrovoc:c_2746
🇫🇷 skos:altLabel = Antenaise
🇬🇧 skos:altLabel = Ewe hoggs
🇬🇧 skos:altLabel = Gimmers
🇫🇷 skos:prefLabel = Brebis
🇬🇧 skos:prefLabel = Ewes

agrovoc:c_4163
🇫🇷 skos:prefLabel = Agneau
🇬🇧 skos:prefLabel = Lambs

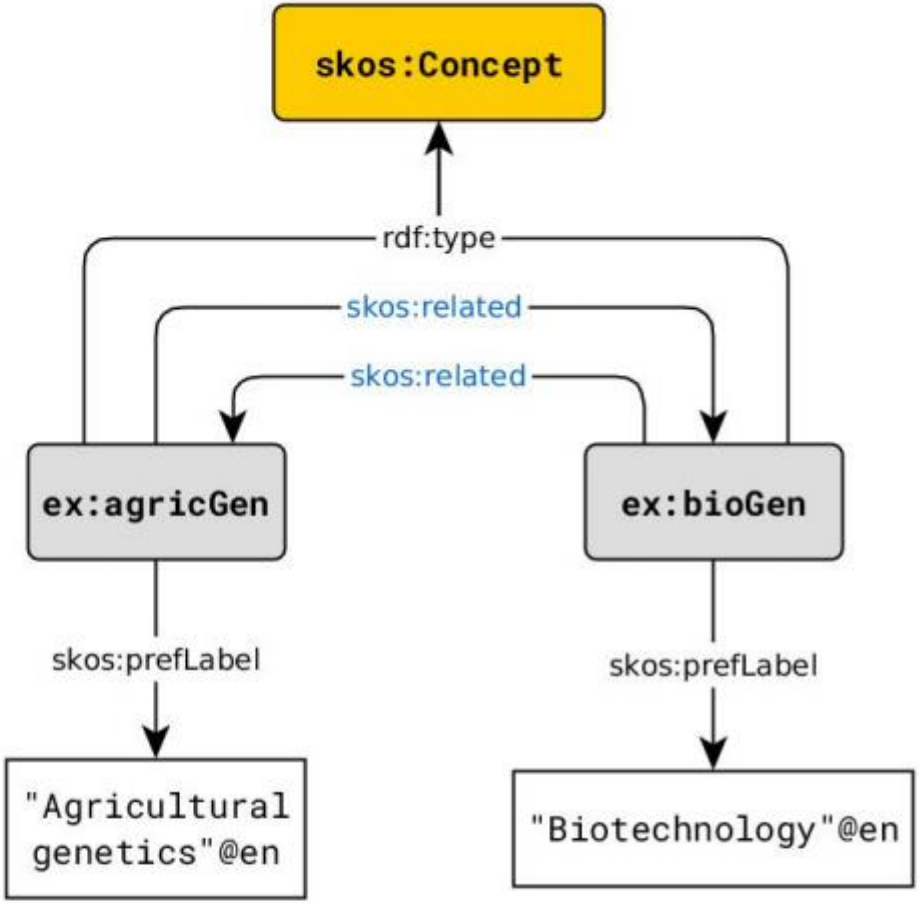
agrovoc:c_8369
🇬🇧 skos:altLabel = Hoggets
🇬🇧 skos:altLabel = Wether hoggs
🇫🇷 skos:prefLabel = Mouton
🇬🇧 skos:prefLabel = Wethers
🇬🇧 skos:scopeNote = Castrated male sheep
🇫🇷 skos:scopeNote = Ovin mâle castré

agrovoc:c_6443
🇬🇧 skos:altLabel = Tups
🇫🇷 skos:prefLabel = Bélier
🇬🇧 skos:prefLabel = Rams

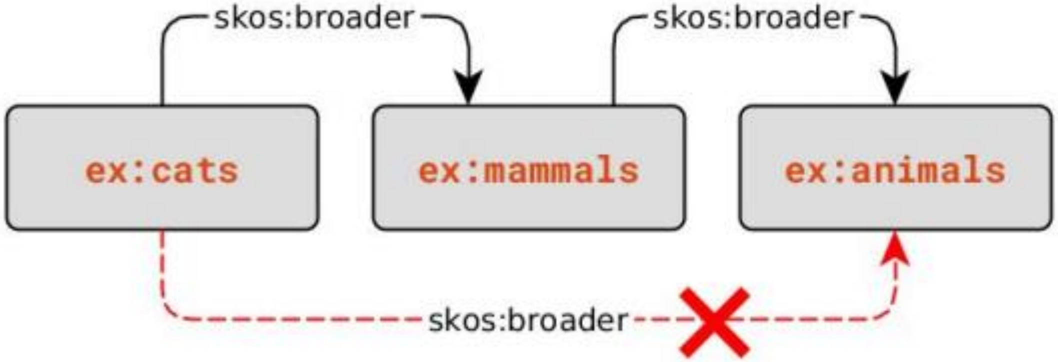
Example (in Turtle)

```
agrovoc:c_4397
a skos:Concept ;
skos:prefLabel "Bétail"@fr , "Livestock"@en ;
skos:altLabel "Animal stock"@en , "Farm animals"@en .
agrovoc:c_2746
a skos:Concept ;
skos:prefLabel "Brebis"@fr , "Ewes"@en ;
skos:altLabel "Gimmers"@en , "Antenaise"@fr , "Ewe hoggs"@en .
agrovoc:c_4163
a skos:Concept ;
skos:prefLabel "Agneau"@fr , "Lambs"@en .
agrovoc:c_6443
a skos:Concept ;
skos:prefLabel "Bélier"@fr , "Rams"@en ;
skos:altLabel "Tups"@en .
agrovoc:c_8369
a skos:Concept ;
skos:prefLabel "Wethers"@en , "Mouton"@fr ;
skos:altLabel "Wether hoggs"@en , "Hoggets"@en ;
skos:scopeNote "Ovin mâle castré"@fr , "Castrated male
sheep"@en .
agrovoc:c_7030
a skos:Concept ;
skos:prefLabel "Sheep"@en , "Ovin"@fr ;
skos:altLabel "Ovis aries"@en , "Ovis aries"@fr ;
skos:broader agrovoc:c_4397 ;
skos:narrower agrovoc:c_2746, agrovoc:c_6443, agrovoc:c_8369,
agrovoc:c_4163 .
```

More semantic relations (1)



More semantic relations (2)



skos:broader

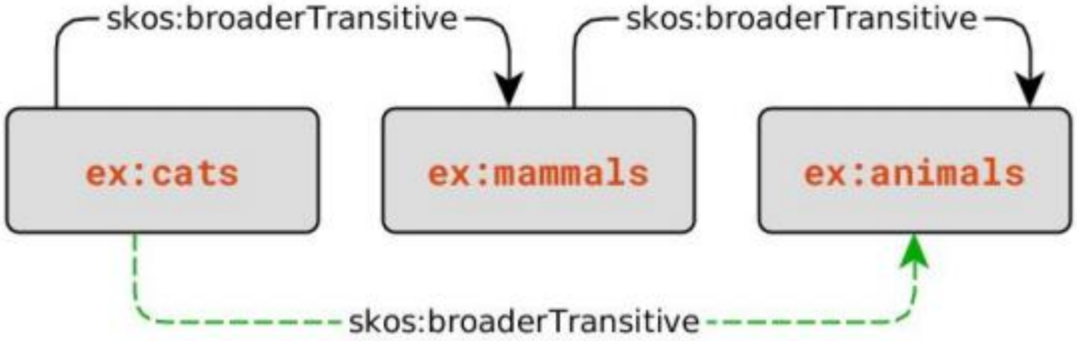
skos:broader

ex:cats

ex:mammals

ex:animals

skos:broader



skos:broaderTransitive

skos:broaderTransitive

ex:cats

ex:mammals

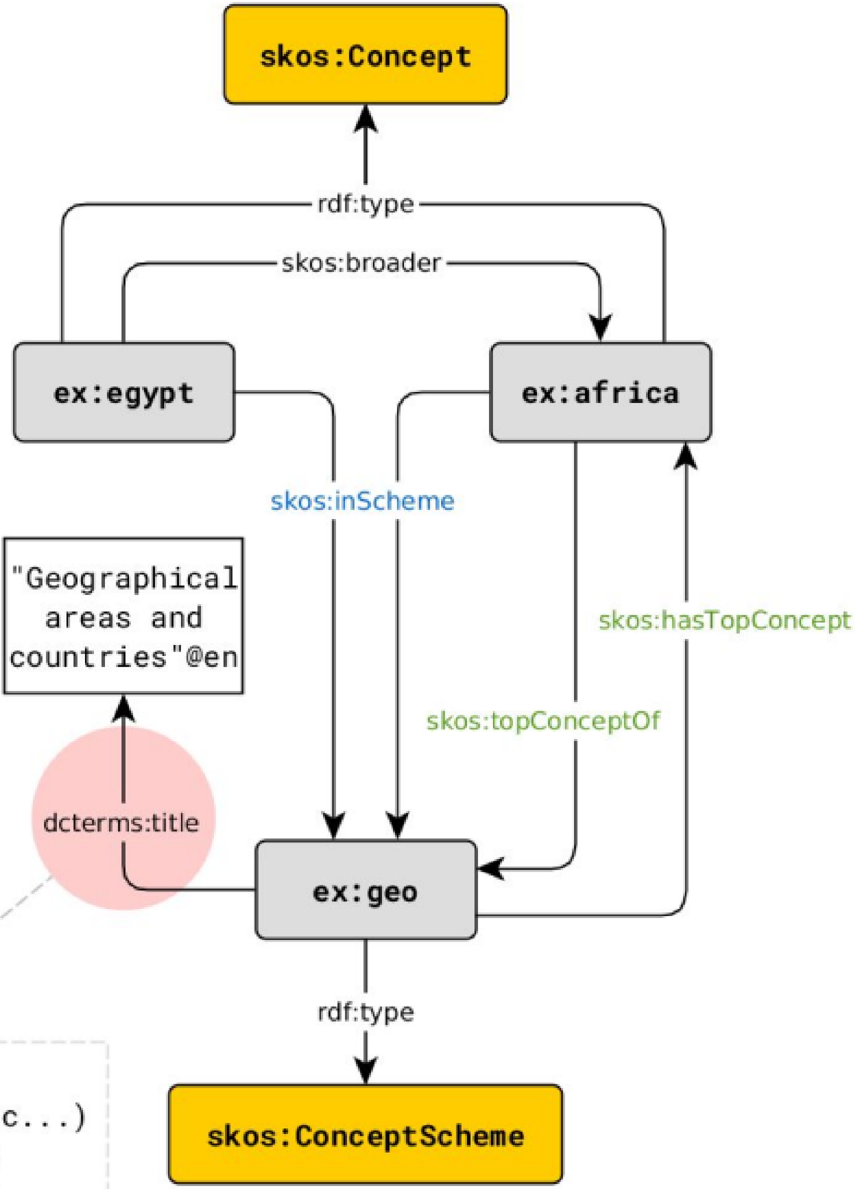
ex:animals

skos:broaderTransitive

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

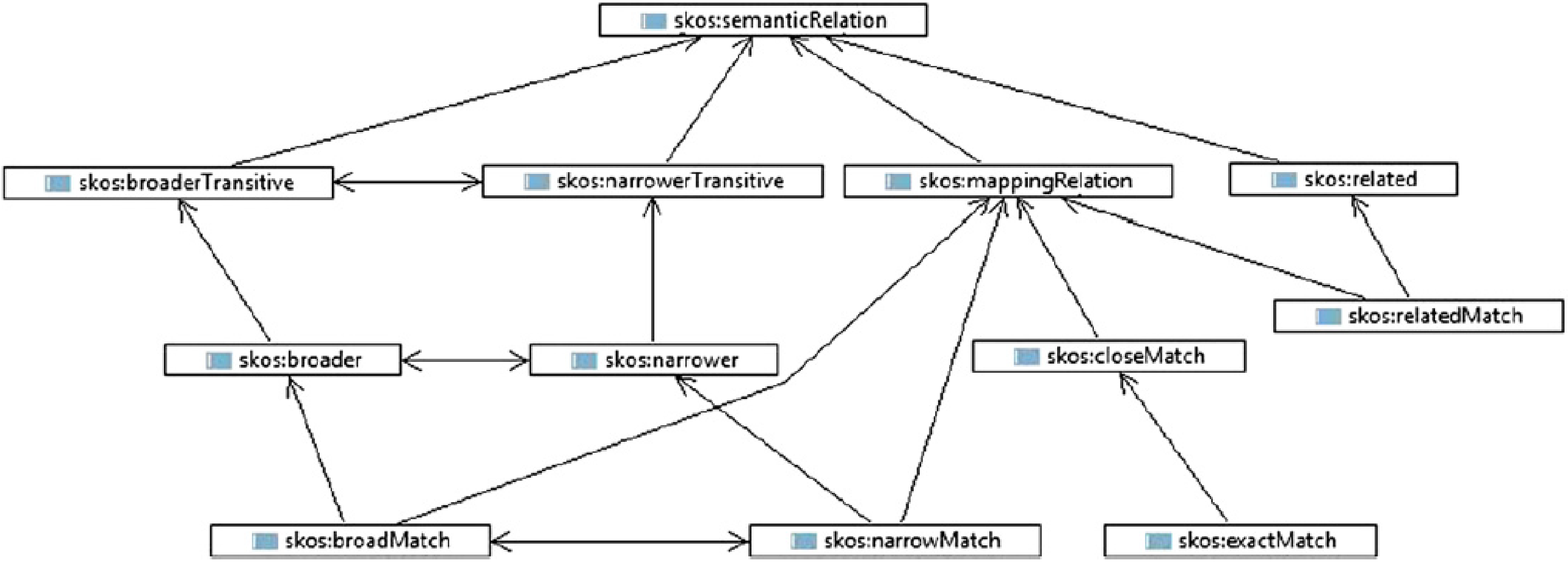
- **skos:hasTopConcept**
  The property is used to link a concept scheme to the concept(s) at the top of the hierarchical structure(s) for that scheme.

- **skos:topConceptOf**
  Inverse of skos:hasTopConcept. Used when a concept is at the top of the hierarchical structure of a concept scheme.



Some details of the concept schemes (title, date of edition, etc...) are usually described with non-SKOS properties (i.e. Dublin Core Terms).

Links between semantic relations



Single arrow: rdfs:subproperty

Double arrow: owl:inverseOf

Advantages

The semantic relations allow any thesaurus to be organised as a hierarchy of concepts

Semantic relations have rules (transitivity, inverse etc.) which allow inferences

E.g. skos:narrower inverse of skos:broader leads to the following triples being *inferred*

```
agrovoc:c_4397 skos:narrower agrovoc:c_7030 .
agrovoc:c_2746 skos:broader agrovoc:c_7030 .
agrovoc:c_4163 skos:broader agrovoc:c_7030 .
agrovoc:c_6443 skos:broader agrovoc:c_7030 .
agrovoc:c_8369 skos:broader agrovoc:c_7030 .
```

RDF Cube

What is Cube?

- A statistical data set comprises a collection of observations made at some points across some logical space.
- The collection can be characterized by a set of dimensions that define what the observation applies to (e.g. time, area, gender) along with metadata describing what has been measured (e.g. economic activity, population), how it was measured and how the observations are expressed (e.g. units, multipliers, status).
- We can think of the statistical data set as a multi-dimensional space, or hyper-cube, indexed by those dimensions.
- This space is commonly referred to as a *cube* for short.
- It is not meant to imply that there are exactly three dimensions (there can be more or fewer) nor that all the dimensions are somehow similar in size.

Example Data Cube
Table 1

	2004-2006		2005-2007		2006-2008	
	Male	**Female**	**Male**	**Female**	**Male**	**Female**
Newport	76.7	80.7	77.1	80.9	77.0	81.5
Cardiff	78.7	83.3	78.6	83.7	78.7	83.4
Monmouthshire	76.6	81.3	76.5	81.5	76.6	81.7
Merthyr Tydfil	75.5	79.1	75.5	79.4	74.9	79.6

*data set extracted from StatsWales report number 003311 which describes life expectancy broken down by region (unitary authority), age and time

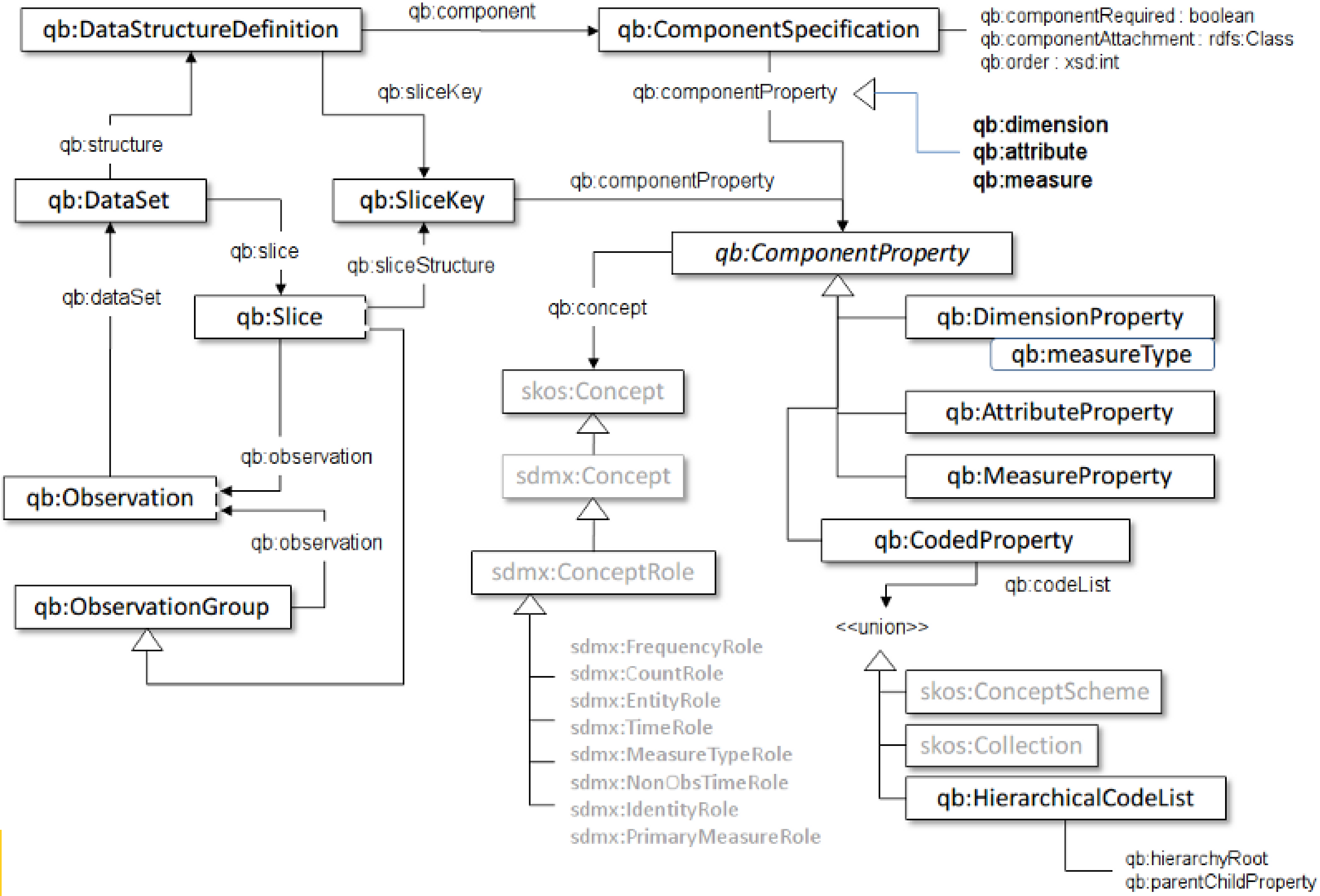
Benefits Of Publishing Multi-dimensional Data As Linked Data Using RDF:

- The individual observations, and groups of observations, become (web) addressable. This allows publishers and third parties to annotate and link to this data; for example a report can reference the specific figures it is based on allowing for fine grained provenance trace-back.
- Data can be flexibly combined across datasets sets (for example *find all Religious schools in census areas with high values for National Indicators pertaining to religious tolerance*). The statistical data becomes an integral part of the broader web of linked data.
- For publishers who currently only offer static files then publishing as linked-data offers a flexible, non-proprietary, machine readable means of publication that supports an out-of-the-box web API for programmatic access.
- It enables reuse of standardized tools and components.

Official Documentation for RDF Data Cube Vocabulary

- Latest document: https://www.w3.org/TR/vocab-data-cube/
- Vocabulary in non-normative format: https://raw.githubusercontent.com/UKGovLD/publishing-statistical-data/master/specs/src/main/vocab/cube.ttl

Key Terms & Relationships



qb:DataStructureDefinition

qb:component

qb:ComponentSpecification

qb:componentRequired : boolean
qb:componentAttachment : rdfs:Class
qb:order : xsd:int

qb:sliceKey

qb:componentProperty

qb:structure

qb:dimension
qb:attribute
qb:measure

qb:DataSet

qb:SliceKey

qb:componentProperty

qb:slice

qb:sliceStructure

qb:ComponentProperty

qb:dataSet

qb:Slice

qb:concept

qb:DimensionProperty

qb:measureType

skos:Concept

qb:AttributeProperty

qb:observation

sdmx:Concept

qb:MeasureProperty

qb:Observation

qb:observation

qb:CodedProperty

sdmx:ConceptRole

qb:codeList

qb:ObservationGroup

<<union>>

sdmx:FrequencyRole
sdmx:CountRole
sdmx:EntityRole
sdmx:TimeRole
sdmx:MeasureTypeRole
sdmx:NonObsTimeRole
sdmx:IdentityRole
sdmx:PrimaryMeasureRole

skos:ConceptScheme

skos:Collection

qb:HierarchicalCodeList

qb:hierarchyRoot
qb:parentChildProperty

Terms

- *DataSet*: A collection of statistical data that corresponds to a defined structure. It may contain observations (actual data/measured values), Organizational structure such as dimensions and slices, structural metadata such as unit of measure and reference metadata such as categorization of the dataset, its publisher, and a SPARQL endpoint where it can be accessed
- *Components*: in a cube are a a set of *dimensions*, *attributes* and *measures*
- *dimension* components serve to identify the observations. A set of values for all the dimension components is sufficient to identify a single observation. Examples of dimensions in table 1 are Gender, Time Period and Region
- *Measure* components represent the phenomenon being observed. “Life Expectancy” is the measure of Table 1.
- *Attribute* components allow us to qualify and interpret the observed value(s). They enable specification of the units of measure, any scaling factors and metadata such as the status of the observation (e.g. *estimated*, *provisional*). “Number of Years” is an attribute for cube represented in Table 1.

Links to other ontologies

SKOS

- Skos:Concept
- Skos:ConceptScheme
- Skos:Collection

SDMX

- Sdmx:ConceptRole
- The SDMX (Statistical Data and Metadata eXchange) standard is highly used as it includes a set of *content oriented guidelines* (COG) which define a set of common statistical concepts and associated code lists that are intended to be reusable across data sets.

qb:DataStructureDefinition

- A qb:DataStructureDefinition defines the structure of one or more datasets, the dimensions, attributes and measures used in the dataset along with qualifying information such as ordering of dimensions and whether attributes are required or optional.
- Advantages of explicit declaration of structure are:
  - enables verification that the data set matches the expected structure
  - helps detecting incoherent sets obtained combining differently structured source data
  - simplifies the data consumption by allowing a consumer to easily determine what dimensions are available for query and their presentational order, which in turn simplifies data consumption
  - supports transmission of the structure information in associated SDMX data flows

Component Properties

- The data structure definition has a number of *components*. In the example table, we have 4 components:
    - Time period
    - Region
    - Sex
    - Life expectancy
- A component can be:
    - A dimension (Time period, region and sex in example table)
    - A measure (life expectancy in example table)
- Each component is an instance of the abstract qb:ComponentProperty class, which in turn has subclasses qb:DimensionProperty, qb:AttributeProperty and qb:MeasureProperty.
- qb:concept: The same *concept* can be manifested in different components. (e.g.- currency can be a dimension or an attribute). To support reuse qb:concept property can be used to link any qb:ComponentProperty to the concept it represents.

Example - Model Table 1 Dimensions

- Three dimensions to represent - time period, region (unitary authority) and sex.

```
eg:refPeriod a rdf:Property, qb:DimensionProperty;
        rdfs:label "reference period"@en;
        rdfs:subPropertyOf sdmx-dimension:refPeriod; #reuse predefined property sdmx-dimension:refPeriod
        rdfs:range interval:Interval; # interval from http://reference.data.gov.uk/def/intervals/
        qb:concept sdmx-concept:refPeriod . #reuse predefined property sdmx-concept:refPeriod

eg:refArea a rdf:Property, qb:DimensionProperty;
        rdfs:label "reference area"@en;
        rdfs:subPropertyOf sdmx-dimension:refArea; #reuse predefined property sdmx-dimension:refArea
        rdfs:range admingeo:UnitaryAuthority; # admingeo:UnitaryAuthority  from http://data.ordnancesurvey.co.uk/ontology/admingeo/
        qb:concept sdmx-concept:refArea . #reuse predefined property sdmx-concept:refArea
```

We can use sdmx-dimension:sex property directly for sex dimension

- Observe the reuse of entities borrowed from different other ontologies to define rdfs:range and rdfs:subPropertyOf without defining everything from scratch.
- Also reuse of entities from RDMX standard

Example - Model Table 1 Measure and Attributes

- A single (primary) measure which corresponds to the topic of the data set - life expectancy

```
eg:lifeExpectancy a rdf:Property, qb:MeasureProperty;
        rdfs:label "life expectancy"@en;
        rdfs:subPropertyOf sdmx-measure:obsValue;
        rdfs:range xsd:decimal .
```

- Measure encodes a value in years.

  Reuse sdmx-attribute:unitMeasure and express value in DBpedia resource http://dbpedia.org/resource/Year

Example - Model Table 1 ComponentSpecifications and DataStructureDefinitions

- To combine measures, attributes and properties we need to declare a qb:DataStructureDefinition resource which in turn will reference a set of qb:ComponentSpecification resources.
- The qb:DataStructureDefinition will be reusable across other data sets with the same structure.
- Other than linking the components to qb:ComponentSpecification using sub-properties, we can declare attributes as optional or required, order components by giving an integer value for qb:order, and attach attributes that are common to overall dataset using qb:componentAttachment property
- Below is the OWL data structure definition of table 1, with id “eg:dsd-le ”

```
 eg:dsd-le a qb:DataStructureDefinition;
       # The dimensions
       qb:component [ qb:dimension eg:refArea;qb:order 1 ];
       qb:component [ qb:dimension eg:refPeriod;qb:order 2 ];
       qb:component [ qb:dimension sdmx-dimension:sex; qb:order 3 ];
       # The measure(s)
       qb:component [ qb:measure eg:lifeExpectancy];
       # The attributes
       qb:component [         qb:attribute sdmx-attribute:unitMeasure;
                             qb:componentRequired "true"^^xsd:boolean;
                             qb:componentAttachment qb:DataSet; ] .
```

Handling Multiple Measures

- When a single observation contain more than one measures, (e.g- if Table 1 contain average height measurement in addition to life expectancy) it can be represented in two alternative approaches

1. **Multi-measure observations** – Declare multiple qb:MeasureProperty components in the data structure definition and attach an instance of each property to the observations within the data set. The limitation is we cannot attach attributes related to individual measures such as the unit of measure
2. ***Measure dimension* –** Define multiple "qb:measure" components at data structure definition and create one observation for each measure, linking particular measure by "qb:measureType"

- Examples can be found at: https://www.w3.org/TR/vocab-data-cube/

Example - Model Table 1 Data Sets

- A resource representing the entire data set is created and typed as qb:DataSet and linked to the corresponding data structure definition via the qb:structure property.

```
eg:dataset-le1 a qb:DataSet;
        rdfs:label "Life expectancy"@en;
        rdfs:comment "Life expectancy within Welsh Unitary authorities - extracted from
        Stats Wales"@en;
        qb:structure eg:dsd-le ;.
```

Example - Model Table 1 Observations

- Each observation is represented as an instance of type qb:Observation.
- In the basic case values for each of the attributes, dimensions and measurements are attached directly to the observation.
- The observation is linked to the containing data set using the qb:dataSet property.

```
eg:o1 a qb:Observation;
  qb:dataSeteg:dataset-le1 ;
  eg:refArea ex-geo:newport_00pr ;
  eg:refPeriod <http://reference.data.gov.uk/id/gregorian-interval/2004-01-01T00:00:00/P3Y>;
  sdmx-dimension:sex sdmx-code:sex-M ;
  sdmx-attribute:unitMeasure <http://dbpedia.org/resource/Year>;
  eg:lifeExpectancy 76.7 ;.
```

- As sdmx-attribute:unitMeasure is repeated for each observation we can include it at the DataSet definition to avoid redundancy.

Representing DataSet Metadata

- DataSets should be marked up with metadata to support discovery, presentation and processing. Dublin Core Terms [DC11] should be used for representing the key metadata annotations commonly needed for DataSets. The RDFS terms for display label (rdfs:label) descriptive comment (rdfs:comment) should be given
- The recommend core set of metadata terms is:
  - dct:title
  - rdfs:label - may be same as dct:title
  - dct:description
  - rdfs:comment - may be same as dct:description
  - dct:issued
  - dct:modified
  - dct:subject
  - dct:publisher
  - dct:license
- Categorizing data sets into different statistical domains, such as *Education*, *Labour*, or *Transportation can be done by using term* dct:subject. The classification schemes are typically represented using the SKOS vocabulary.
- For convenience the SMDX Subject-matter Domains have been encoded as a SKOS concept scheme at http://purl.org/linked-data/sdmx/2009/subject#. Thus our sample dataset might be marked up as:

```
eg:dataset1 a qb:DataSet;rdfs:label"Life expectancy"@en;
dct:title"Life expectancy"@en;
rdfs:comment"Life expectancy within Welsh Unitary authorities - extracted from Stats Wales"@en;
dct:description "Life expectancy within Welsh Unitary authorities - extracted from Stats Wales"@en;
dct:issued"2010-08-11"^^xsd:date;
dct:subject
sdmx-subject:3.2 ,# regional and small area statistics
sdmx-subject:1.4 ,# Health
eg:Wales;# Wales, a custom skos:Concept
```

- Publisher should be e recorded as part of the dataset metadata. Recommended term is dct:publisher. The organization can be represented as an instance of foaf:Agent, or some more specific subclass such as org:Organization.

```
eg:dataset1 a qb:DataSet;
            dct:publisher <http://example.com/meta#organization>.
```

Example Implemented

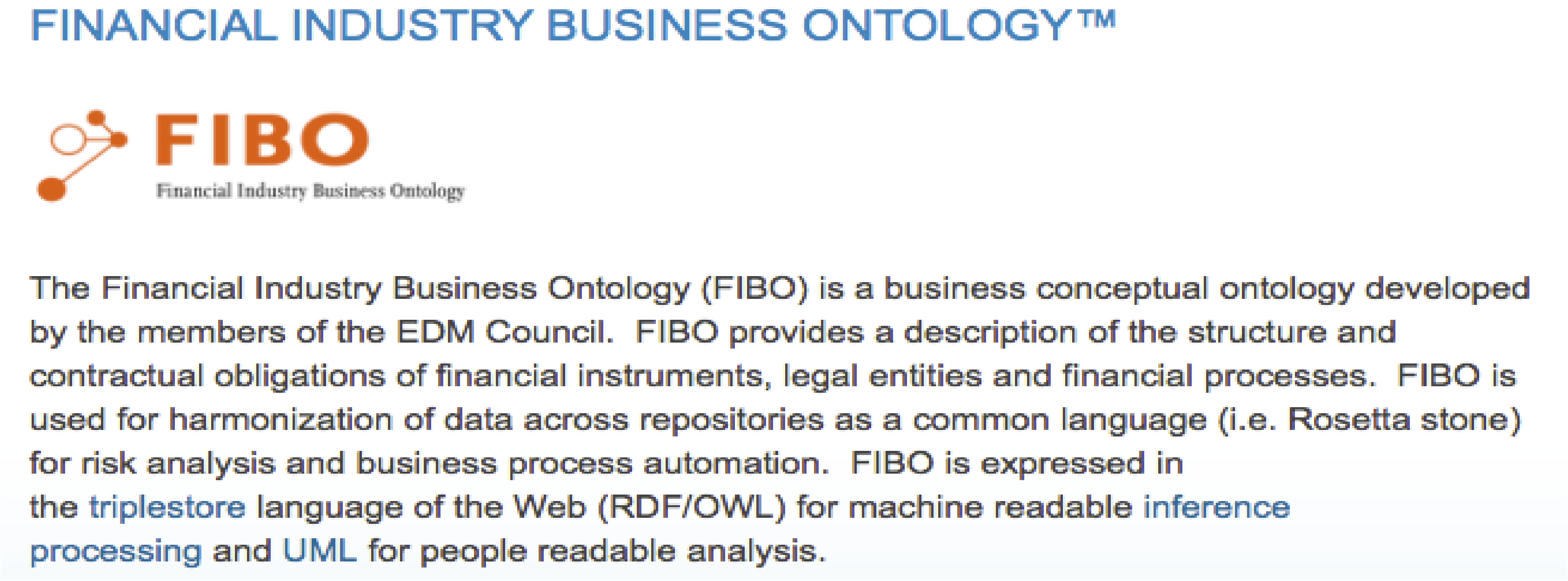
- You can find the complete implementation of Table 1 in following URL:

http://adage2.cse.unsw.edu.au/Resources/SampleData/TripleData/LifeExpectancyDataSet.ttl

- You can import it to Protégé and explore

Financial Industry Business Ontology (FIBO)

FIBO



FINANCIAL INDUSTRY BUSINESS ONTOLOGY™

The Financial Industry Business Ontology (FIBO) is a business conceptual ontology developed by the members of the EDM Council. FIBO provides a description of the structure and contractual obligations of financial instruments, legal entities and financial processes. FIBO is used for harmonization of data across repositories as a common language (i.e. Rosetta stone) for risk analysis and business process automation. FIBO is expressed in the triplestore language of the Web (RDF/OWL) for machine readable inference processing and UML for people readable analysis.

FIBO will contribute to transparency in the global financial system, aid industry firms in providing a cost-effective means for integrating disparate technical systems and message formats, and aid in regulatory reporting by providing clear and unambiguous meaning of data from authoritative sources.

http://www.edmcouncil.org/financialbusiness

Four Elements of FIBO

-FIBO Foundations

http://www.omg.org/spec/EDMC-FIBO/FND/

-FIBO Business Entities

http://www.omg.org/spec/EDMC-FIBO/BE/

-FIBO Indices and Indicators

http://www.omg.org/spec/EDMC-FIBO/IND/

-Financial Instrument Global Identifier http://www.omg.org/spec/FIGI/

Some of these are still developing, and it is possible for new concepts to be introduced over time.

Fundamentals - FIBO Foundation

Latest version: http://www.omg.org/spec/EDMC-FIBO/FND/1.1/Beta2/index.htm

- Presents a pdf document with descriptions on each concept of Foundation.

http://www.omg.org/spec/EDMC-FIBO/FND/1.1/Beta2/PDF/index.htm

- Each of the concept is represented in a machine consumable .rdf file and UML .xmi file as well
  - E.g.: Concept of Agreement
    - http://www.omg.org/spec/EDMC-FIBO/FND/20160201/Agreements/Agreements.rdf
    - http://www.omg.org/spec/EDMC-FIBO/FND/20160201/Agreements/Agreements.xml

Useful Resources

FIBO in Semantic World

1. https://www.brighttalk.com/webcast/9291/102997
2. http://www.omg.org/news/meetings/workshops/FS-CONF-pres/1300_Newman.pdf

Thank You