Schemas and Vocabularies – RDFS,SKOS in detail, OWL basics

Bogdan Kostov, Miroslav Blaško

bogdan.kostov@fel.cvut.cz, miroslav.blasko@fel.cvut.cz

October 26, 2017

Outline

Introduction

What is Data?

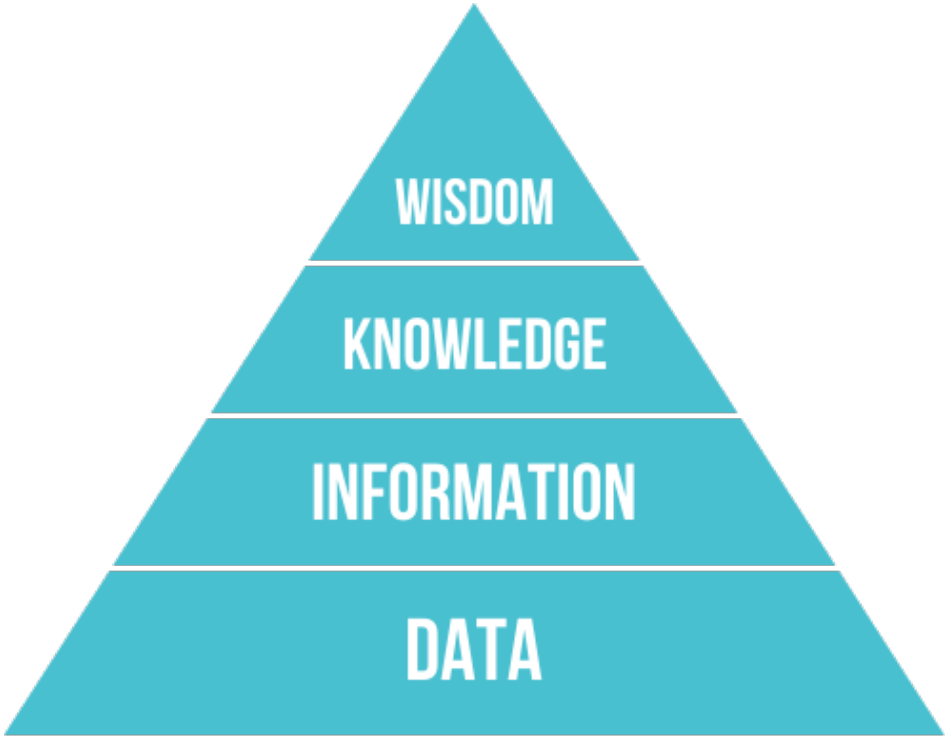


Figure: Data, Information, Knowledge and Wisdom[1]

[1] Image source wikipedia.org/wiki/DIKW_pyramid

Conceptual, logical and physical schemas

- Conceptual schema - describes the semantics of a domain
- Logical schema - describes the structure of the data
- Physical schema - describes the physical means used to store data

Semantics and Schemas

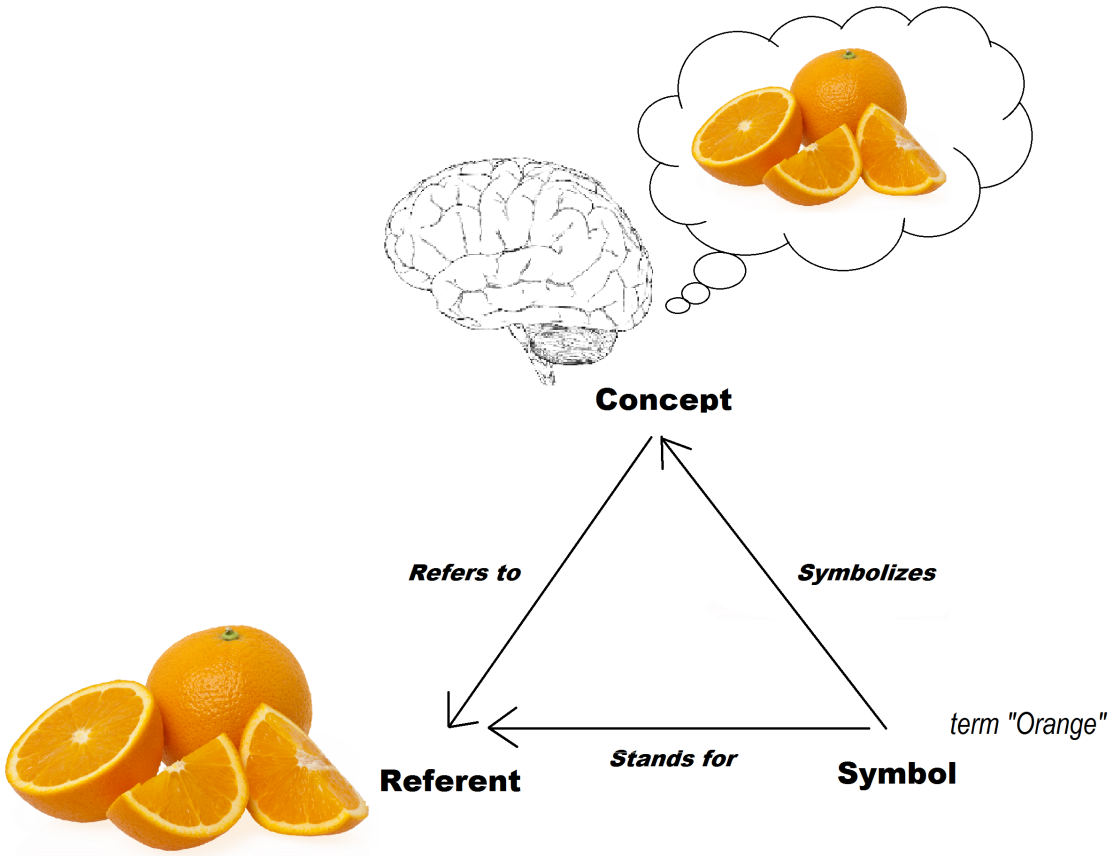


Figure: The meaning triangle according to Ogden&Richards, 1969

Semantics and Schemas II

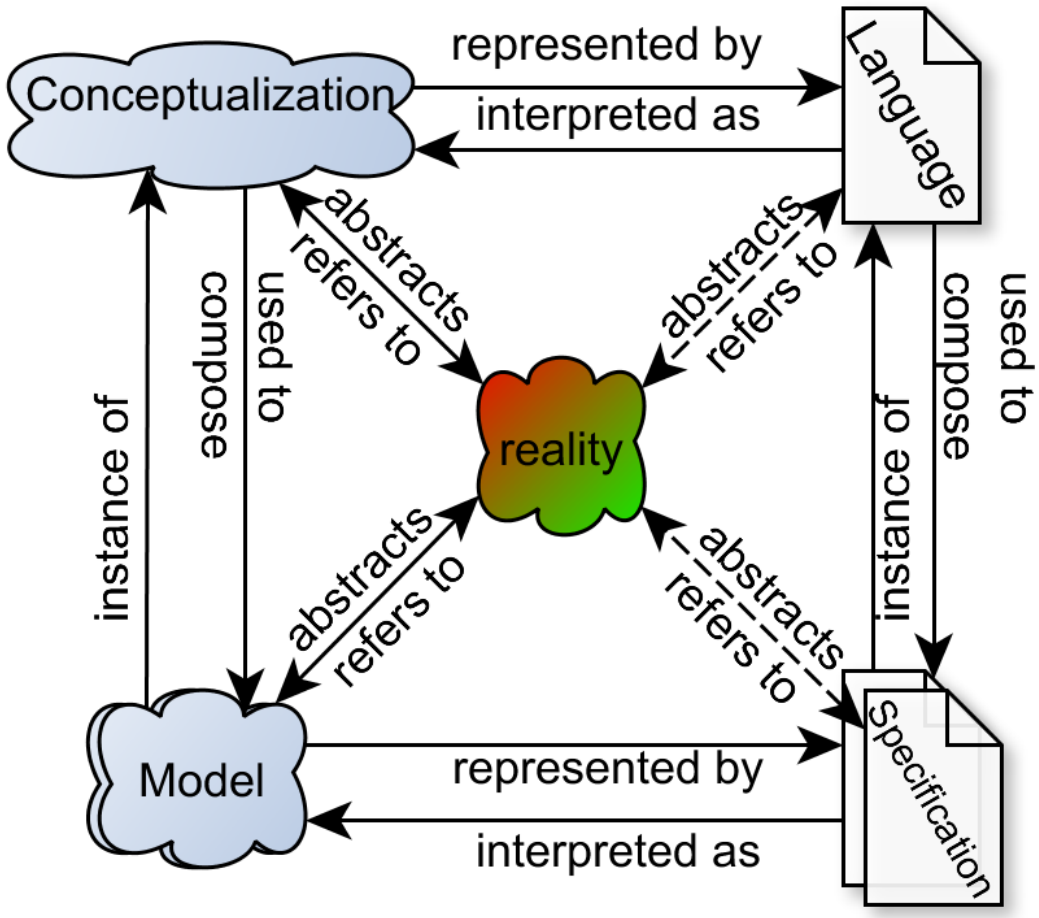


Figure: Relations between conceptualization, Model, Modeling Language and Specificationvariation [2]

[2] Adopted from [**Guizzardi2005**]

Terminology

- Model - Structured Data (data), e.g., RDF
- Language - Ontology, Schema, Taxonomy, Vocabulary

Terminology

- Model - Structured Data (data), e.g., RDF
- **Language - Ontology, Schema, Taxonomy, Vocabulary**

How to represent Ontologies

Ontology Engineering

RDFS Simple Metamodeling Language

- Ontologies are complex
- Ontologies are hard to develop
- Ontologies often evolve

Practical constraints force different levels of description of Ontologies. We will discuss several LD compliant methods of describing Ontologies:

- RDFS
- OWL
- SKOS

Metamodeling in RDFS

RDFS Simple Metamodeling Language

- basic metamodeling vocabulary for the description of classes and properties

```
rdf:type,
rdfs:Class,
rdfs:subClassOf,
rdf:Property,
rdfs:subPropertyOf,
rdfs:domain,
rdfs:range
```

Classes

- define instances :

```
ex:John rdf:type ex:Person .
```

Classes

- define instances :

```
ex:John rdf:type ex:Person .
```

- define classes (class `rdfs:Class`) :

```
ex:Person rdf:type rdfs:Class .
```

Classes

- define instances :

```
ex:John rdf:type ex:Person .
```

- define classes (class `rdfs:Class`) :

```
ex:Person rdf:type rdfs:Class .
```

- create class hierarchies (property `rdfs:subClassOf`) :

```
ex:Woman rdfs:subClassOf ex:Person .
```

Classes

- define instances :

```
ex:John rdf:type ex:Person .
```

- define classes (class `rdfs:Class`) :

```
ex:Person rdf:type rdfs:Class .
```

- create class hierarchies (property `rdfs:subClassOf`) :

```
ex:Woman rdfs:subClassOf ex:Person .
```

- multiple inheritance :

```
ex:Woman rdfs:subClassOf ex:Person .
ex:Woman rdfs:subClassOf ex:Female.
```

Properties

- property definitions (resource `rdf:Property`) :

```
ex:hasParent rdf:type rdf:Property .
```

- creation of property hierarchies (property `rdfs:subPropertyOf`) :

```
ex:hasMother rdfs:subPropertyOf ex:hasParent.
```

- multiple inheritance
- domain and range definition :

```
ex:hasMother rdfs:domain ex:Person .
ex:hasMother rdfs:range ex:Woman
```

- domains/ranges considered as conjunction :

```
ex:hasMother rdfs:range ex:Person .
ex:hasMother rdfs:range ex:Female .
```

RDFS Model – Axiomatic Triples

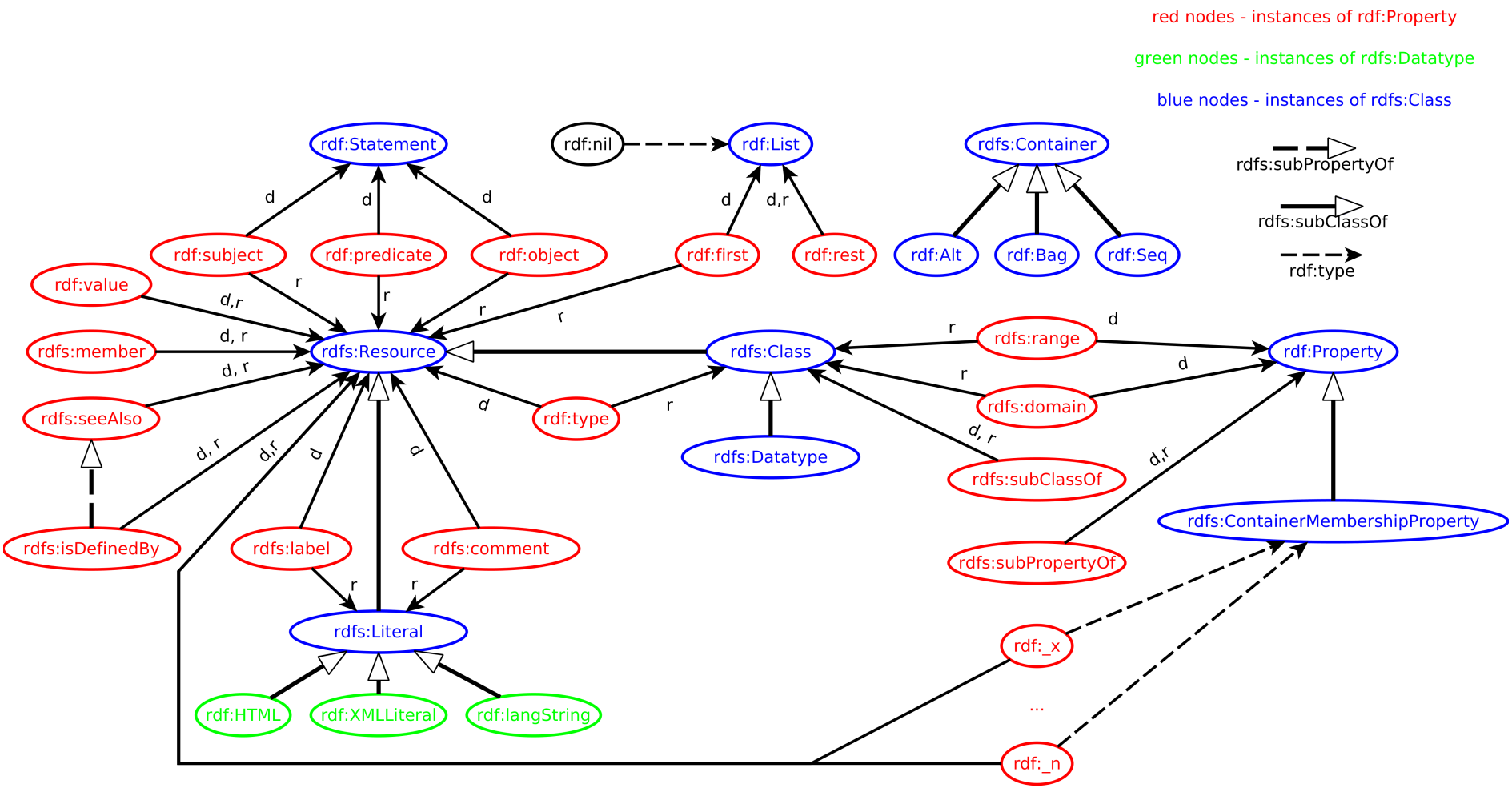


Figure: Visualization of axiomatic triples of RDFS. Precise definition can be found in [**Patel-Schneider:14:RS**]

Metamodeling in OWL 2

Properties

Property Expressions

OWL 2 supports no data property expressions and the only object property expresion:

`inverse` means an inverse property (i.e. property going in the opposite direction),

```
inverse :hasChild
```

Inverse properties can be used in class frames, property frames as well as individuals frames.

Object Property Frames

```
ObjectProperty: :hasMother
    Characteristics: Functional, Irreflexive, Asymmetric
    Domain: :Person
    Range: :Woman
    SubPropertyOf: :hasParent
    EquivalentTo: inverse :isMotherOf
    DisjointWith: :hasFather
    InverseOf: :isMotherOf
    SubPropertyChain: :hasFather o :isWifeOf
```

`Characteristics` – selection of `Functional`,`InverseFunctional`, `Transitive`, `Reflexive`, `Irreflexive`, `Symmetric`, `Asymmetric` – interpreted in their mathematical sense

`Domain`,`Range` have the same meaning as in RDFS

`SubPropertyOf` specifies props representing supersets of the frame property

`EquivalentTo` specifies props semantically equivalent to the frame class

`DisjointWith` specifies props disjoint with the frame property

Data Property Frames

```
DataProperty: :hasBirthNumber
    Characteristics: Functional
    Domain: :Person
    Range: xsd:string
    SubPropertyOf: :hasIdentifyingNumber
```

The only **Characteristics** available is `Functional`. Other sections have the same meaning

Class Expressions

Boolean operators

OWL 2 supports many class modeling constructs including boolean connectives, individual enumeration, and object/data value restrictions.

`owl:Thing`, `owl:Nothing` are two predefined OWL classes containing all (resp. no) individuals,

Boolean operators

OWL 2 supports many class modeling constructs including boolean connectives, individual enumeration, and object/data value restrictions.

`owl:Thing`, `owl:Nothing` are two predefined OWL classes containing all (resp. no) individuals,

`and`,`or`,`not` have the meaning of standard set intersection, union and complement,

```
(:FlyingObject and not :Bat) or :Pinguin
```

Boolean operators

OWL 2 supports many class modeling constructs including boolean connectives, individual enumeration, and object/data value restrictions.

`owl:Thing`, `owl:Nothing` are two predefined OWL classes containing all (resp. no) individuals,

`and`,`or`,`not` have the meaning of standard set intersection, union and complement,

```
(:FlyingObject and not :Bat) or :Pinguin
```

individual enumeration lists individuals belonging to a class expression.

```
{:John :Mary}
```

Object value Restrictions (1)

existential quantification says that a property filler exists (not necessarily in data !)

Listing 1 : A set of objects having at least one son

```
:hasChild some :Man
```

Object value Restrictions (1)

existential quantification says that a property filler exists (not necessarily in data !)

Listing 4 : A set of objects having at least one son

```
:hasChild some :Man
```

universal quantification says that each property filler belongs to a class

Listing 5 : A set of objects having no child or only sons

```
:hasChild only :Man
```

Object value Restrictions (1)

existential quantification says that a property filler exists (not necessarily in data !)

Listing 7 : A set of objects having at least one son

```
:hasChild some :Man
```

universal quantification says that each property filler belongs to a class

Listing 8 : A set of objects having no child or only sons

```
:hasChild only :Man
```

cardinality restriction restricts the number of property fillers

Listing 9 : Sets of objects exactly two (min four/max one) wheels

```
:hasPart exactly 2 :Wheel
:hasPart min 4 :Wheel
:hasPart max 1 :Wheel
```

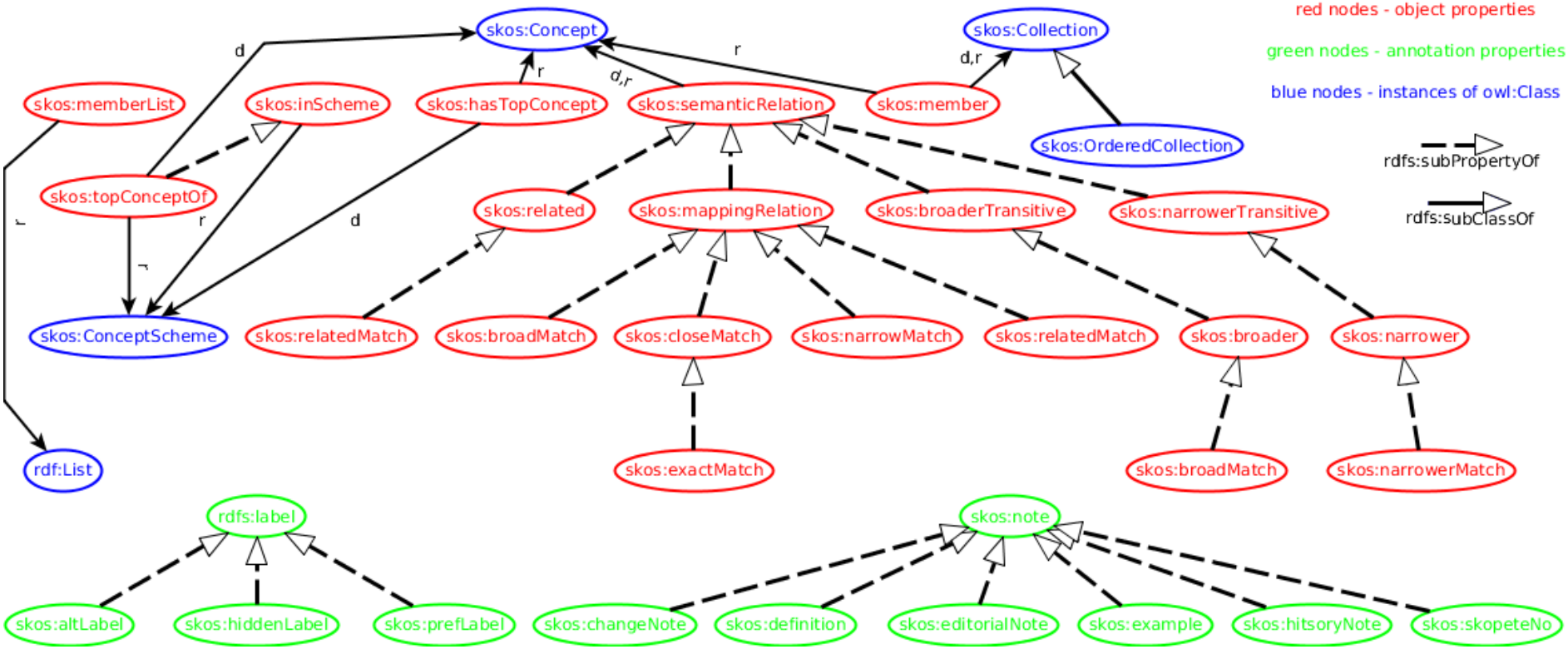
SKOS

What is SKOS ?

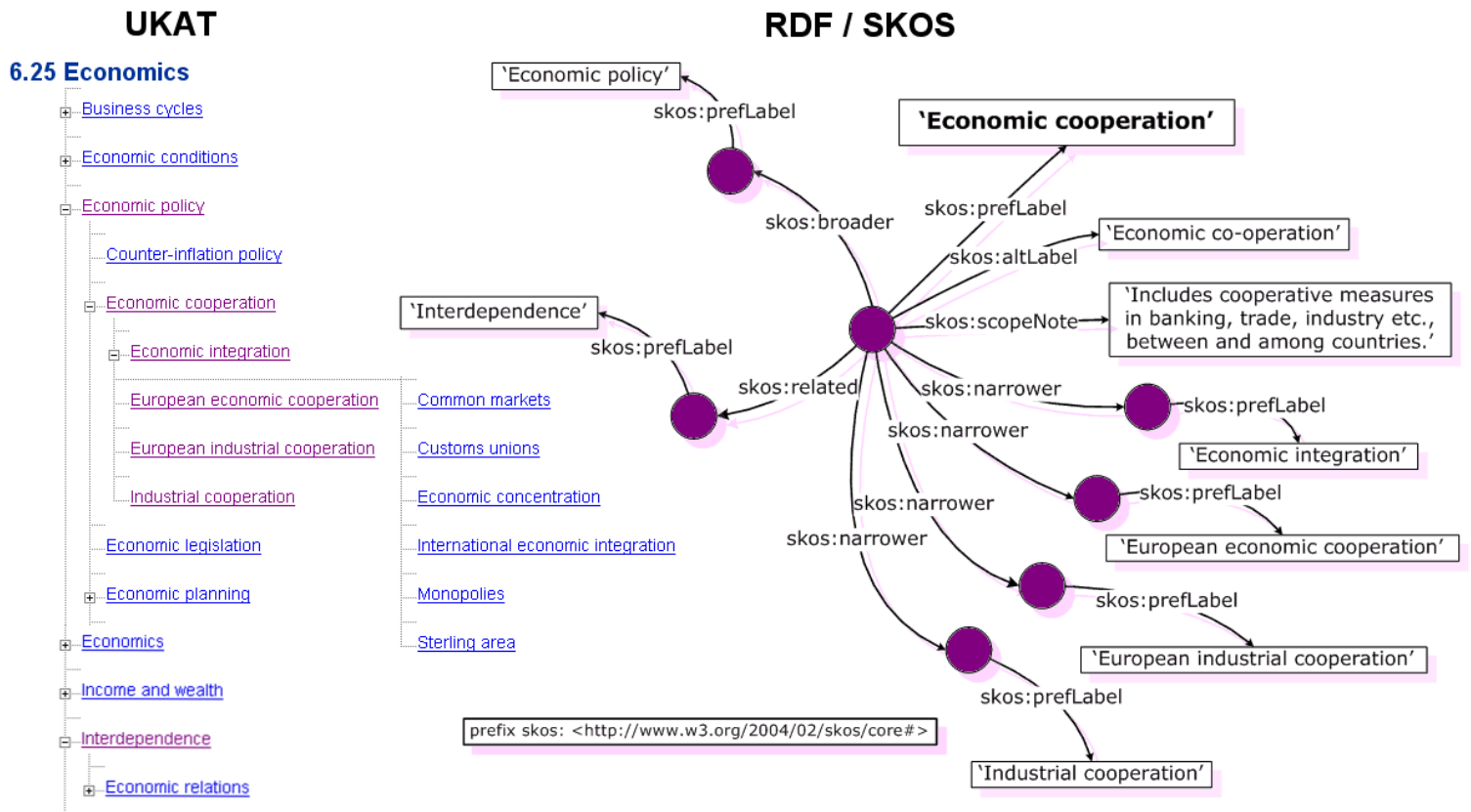
SKOS = **S**imple **K**nowledge **O**rganization **S**ystems

- knowledge organization system that become official W3C recommendation in August 2009
- it allows to express thesauri, classification systems, subject headings, lists ..
- it represent KOS in RDF in simple way
- SKOS data model is formally defined as an OWL Full ontology

Partial SKOS Model – Axiomatic Triples



Example of SKOS representing thesaurus



Portion of UK Archival Thesaurus (UKAT) and its relevant representation in SKOS taken from `http://www.mkbergman.com/374/an-intrepid-guide-to-ontologies/`.

SKOS relevant materials

- SKOS homepage
  `http://www.w3.org/2004/02/skos/`
- SKOS Primer
  `http://www.w3.org/TR/skos-primer/`
- SKOS Reference
  `http://www.w3.org/TR/skos-reference/`
- SKOS Use Cases and Requirements
  `http://www.w3.org/TR/2009/NOTE-skos-ucr-20090818/`

Knowledge Organization Systems for Semantic Web

Knowledge Organization Systems

Why do we need to organize knowledge ?

- Indexing
- Retrieval
- Organization and navigation

Basics

About KOS

KOS = **K**nowledge **O**rganization **S**ystems

- a generic term used to cover all types of schemes for organizing information and promoting knowledge management
- includes following subtypes [**hodge2000systems**] :
  - **Term lists** (authority files, glossaries, dictionaries, gazetteers)
  - **Classification and categories** (subject headings, classification schemes, taxonomies, categorization schemes)
  - **Relationship lists** (thesauri, semantic networks, ontologies)
- another classification [**hedden2010accidental**] defines **Controlled vocabularies** which include all types of KOS except of highly structured semantic networks and ontologies

Types of KOS

What is Controlled vocabulary ?

- **Controlled vocabulary** in its simplest form is a restricted list of words or terms for some specialized purpose
- it is mostly used for descriptive cataloging, tagging or indexing
- "controlled" is used because
  - only terms from the list may be used for the subject area covered
  - only under certain specific conditions may controlled vocabulary change and grow which is responsibility of controlled vocabulary editor, not user

Term lists – Authority file

- **Authority file** is a list of terms that are used to control the variant names for an entity or the domain value for a particular field (names for countries, individuals, and organizations)
- properties
  - might include non/preferred terms
  - generally does not include deep organization or complex structure
- examples
  - the Library of Congress Name Authority File
  - the Getty Geographic Authority File

Term lists – Glossary

- **Glossary** is a list of terms, usually with definitions
- properties
  - the terms may be from a specific subject field or from a particular work
  - the terms are defined within a specific environment and rarely include variant meanings
- examples
  - the Environmental Protection Agency (EPA) Terms of the Environment

Term lists – Dictionary

- **Dictionary** is an alphabetical list of words and their definitions
- properties
  - might include variant senses where applicable
  - more general in scope than glossary
  - might provide synonyms and related words (through the definitions) but it is not explicitly structured or grouped by concept

Term lists – Gazetteer

- **Gazetteer** is a list of place names
- properties
  - each entry may also be identified by feature type (e.g. river, city, or school)
  - geospatially referenced gazetteer provides coordinates for locating the place
- examples
  - the U.S. Code of Geographic Names

Classification and categories – Subject Heading

- **Subject Heading** is a scheme type providing a list of controlled terms to represent the subjects of items in a collection
- properties
  - the list of terms can be extensive and cover a broad range of subjects
  - the list's structure is generally very shallow
- examples
  - the Medical Subject Headings (MeSH)
  - the Library of Congress Subject Headings (LCSH)

Title:	Anti-doping : the fight against ***performance-enhancing drugs*** in sport / [European Commission, Directorate-General for Research].
Publisher:	Luxenbourg : Office for Official Publications of the European Communities, [2001?]
Description:	1 sheet (folded) : col. ill. ; 21 cm.
Series:	European research in action
Portion of title:	Fight against ***performance-enhancing drugs*** in sport
Notes:	Cover title.
	"KI-38-01-601-EN-D."
OCLC:	(OCoLC)ocm48580907
Subjects:	Athletes--Drug use.
	Doping in sports.
Other:	European Commission. Directorate-General for Research.

Example of LCSH for performance enhancing drugs (see "Athletes–Drug" and "Doping in sports") taken from `http://http://madcat.library.wisc.edu/`

Classification and categories – Taxonomy

- **Taxonomy** is collection of terms organized into a hierarchical structure
- properties
  - each term in a taxonomy is in one or more parent-child relationships to other terms in the taxonomy
  - parent-child relationship in a taxonomy can have different parent-child relationships in a taxonomy (e.g., whole-part, genus-species, type-instance)
  - generally each parent-child relationships have a single parent of the same type (otherwise it is called poly-hierarchy)
  - traditional taxonomies use "is a kind of" relationship

Note

- the term "taxonomy" is sometimes used in broader sense to refer any means of organizing concepts of knowledge. To limit it to hierarchical structure within this context term "hierarchical taxonomy" is used.
- we will use this narrower sense when talking about taxonomies.

Relationship lists – Thesaurus

- **Thesaurus** is a list of controlled terms and relationships among them. The terms are organized in groups of synonyms representing concepts.
- properties
  - relationships commonly found in thesaurus include hierarchy (broader/narrower term), equivalence (synonym), and association or relatedness (related term).
  - dictionary-thesaurus includes all the associated terms that can be potentially used in place of term entry in various contexts
  - information retrieval thesaurus on the other hand is used in all contexts within covered domain of interest

ABSORPTION

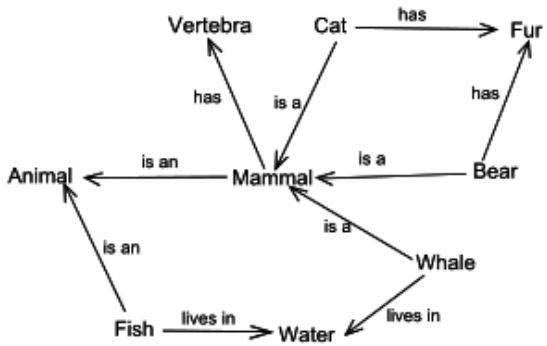
The retention and conversion into another form of energy of rays, waves, or particles by a substance.

UF ABSORPTIVE PROPERTIES
BT **SORPTION**
NT **BIOLOGICAL ABSORPTION**
RESONANCE ABSORPTION
TWO PHOTON ABSORPTION
X RAY ABSORPTION ANALYSIS

Representation of term "absorption" taken from DTIC Thesaurus

Relationship lists – Semantic network

- **Semantic network** is organization of terms and concepts in a structure not as hierarchy but as a network (web)
- properties
  - the concepts are represented as nodes of the network while relationships are edges that connects them
  - the relationships are generally richer than in thesaurus. They may include specific whole-part, cause-efect, or parent-child relationships.
- examples
  - Princeton University's WordNet



Example of simple semantic network about Mammals taken from

`http://en.wikipedia.org/wiki/Semantic_network`

Relationship lists – Ontology

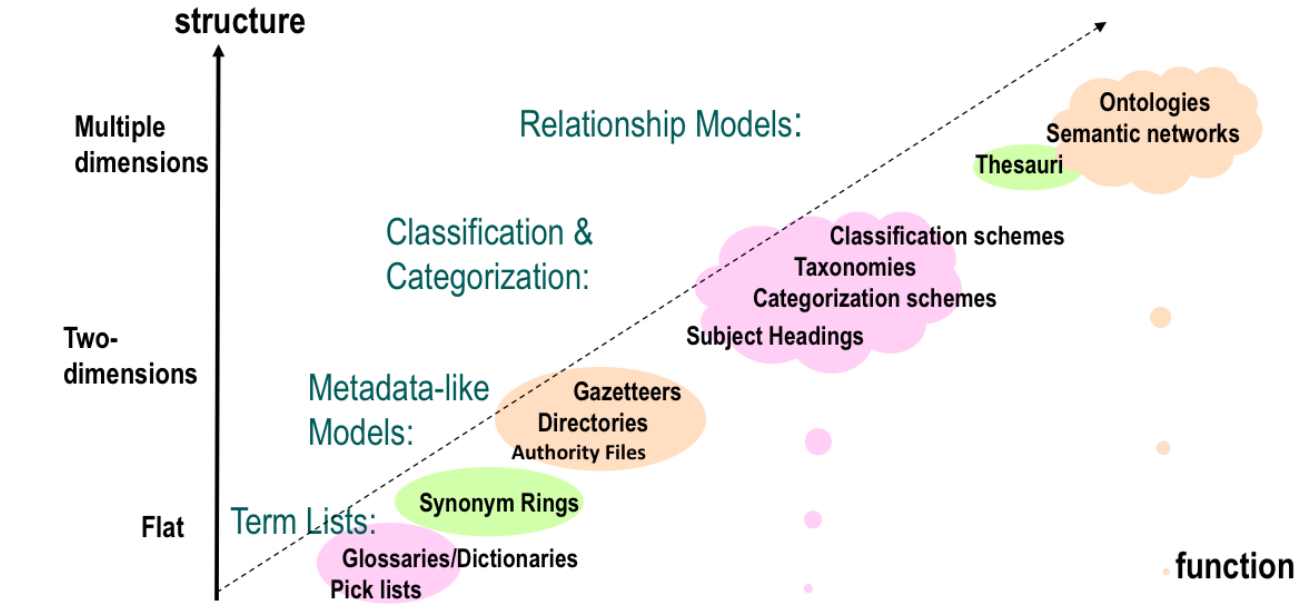
- **Ontology** is structural framework for organizing knowledge that uses controlled vocabulary expressed in an ontology representation language
- properties
  - the language has a grammar for using vocabulary terms to express something meaningful within a specified domain of interest
  - the grammar specifies formal constraints on how terms in the ontology's controlled vocabulary can be used together
  - concepts and relations are organized into "is kind of" hierarchies
  - compared to semantic network it can has axioms and rules
- examples
  - Descriptive Ontology for Linguistic and Cognitive Engineering
  - Basic Formal Ontology

Note

By the term "ontology" we refer here mainly to "formal ontology", i.e. ontology with main purpose to negotiate meaning.

Various types of KOS

Zeng 2008 p.161



Major functions							
eliminating ambiguity	xxx		xxx	xx	xxxx	xx
controlling synonyms		xxxx	xxx	xx	xxxx	xx
establishing relationships: hierarchical			x	xxxx	xxx	xxx
establishing relationships: associative					xxxx	xxxxx
presenting properties						xxxxx

Figure taken from [**lei2008knowledge**] visualize types of KOS, arranged according to the degree of controls introduced (from natural language to controlled language) and the strength of their semantic structure (from weakly structured to strongly structured), corresponding to the major functions of KOS.