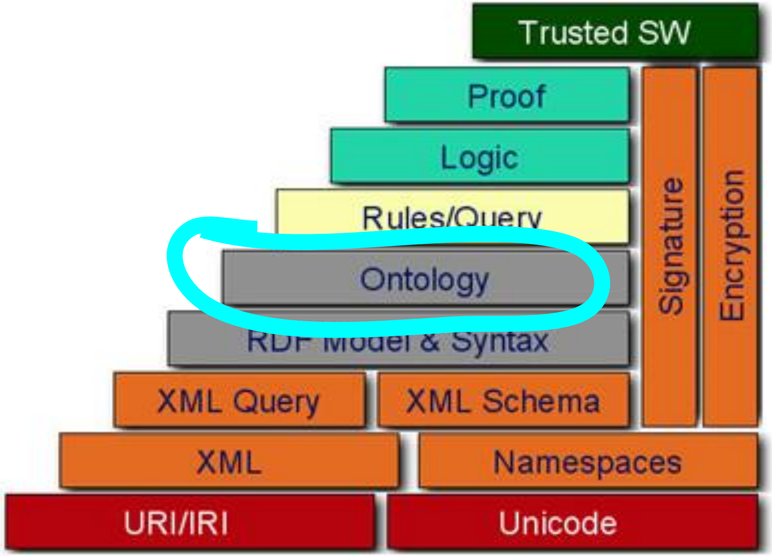
Linguistic Linked Open Data for Humanists

Anas Fahad Khan, Giulia Pedonese, Michele Mallia

Lisbon Summer School in Linguistics, July 1-5, 2024

Funded by the European Union NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Consiglio Nazionale delle Ricerche

Exploring the Semantic Web Stack

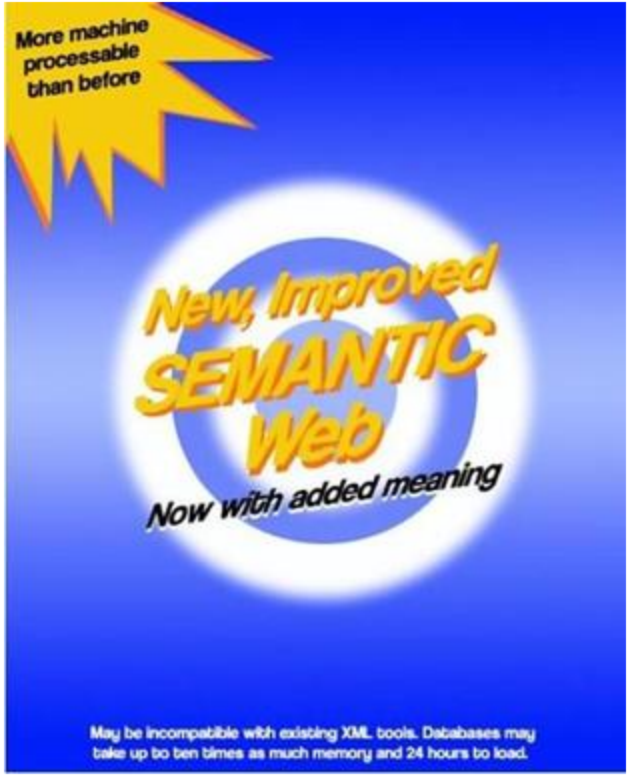


The Semantics in the Semantic Web:

An Interlude on Semantic Artifacts

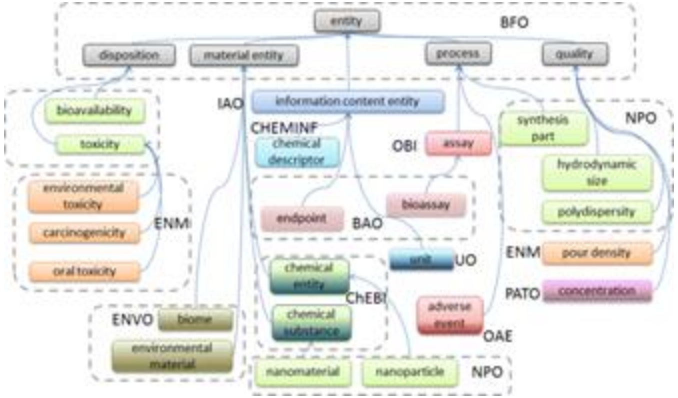
The Semantics of the Semantic Web

- One of the main motivations behind the **Semantic Web** is to publish data on the World Wide Web in a way that makes its **meaning as explicit as possible**: making it easier for machines to carry out tasks on the basis of this meaning **(machine actionability)**.
- One way this is done is via the use of formal languages, **RDF**, **RDFS**, and **OWL** which have a precise mathematical semantics to describe data
- Such languages offer us a means of **representing knowledge** in a way that is **less ambiguous** than natural language (as we have seen with RDF(S)) and that we can reason about in a fairly efficient way using automated reasoning algorithms (without having to use LLMs!).



The Semantics of the Semantic Web

- We can use these formal languages to describe the most important concepts in a domain, via reference **models/conceptualisations,** these machine actionable descriptions can then be published on the Semantic Web and used by others
- We group these models/conceptualisations under the umbrella term of **semantic artifacts**
- Classes and properties from these semantic artifacts can then be re-used via Semantic Web standards
- These semantic artifacts are **especially useful for resource metadata**



Semantic Artifacts

- Term introduced in the **FAIRsFAIR project**
  - “**Machine actionable formalisation** *(represented using appropriate formats and serialisations, including RDF and non-RDF standards) of a* **conceptualisation***, enabling sharing and reuse by* **humans and machines**” (Corcho et al)
- In other words SA’s represent **the conceptual model(s)** used to organise a (structured) dataset, but in a form that’s **shareable** and **accessible** both to computers and to human beings
  - Related term used in the past: **Knowledge Organisation System** (the last part of the name of SKOS!)
- **SA’s** can range from **simple fixed lists of terms** (with definitions), to **thesauri, taxonomies and ontologies**

Kinds of Semantic Artifacts

Strong Semantics

Ontology

Thesaurus

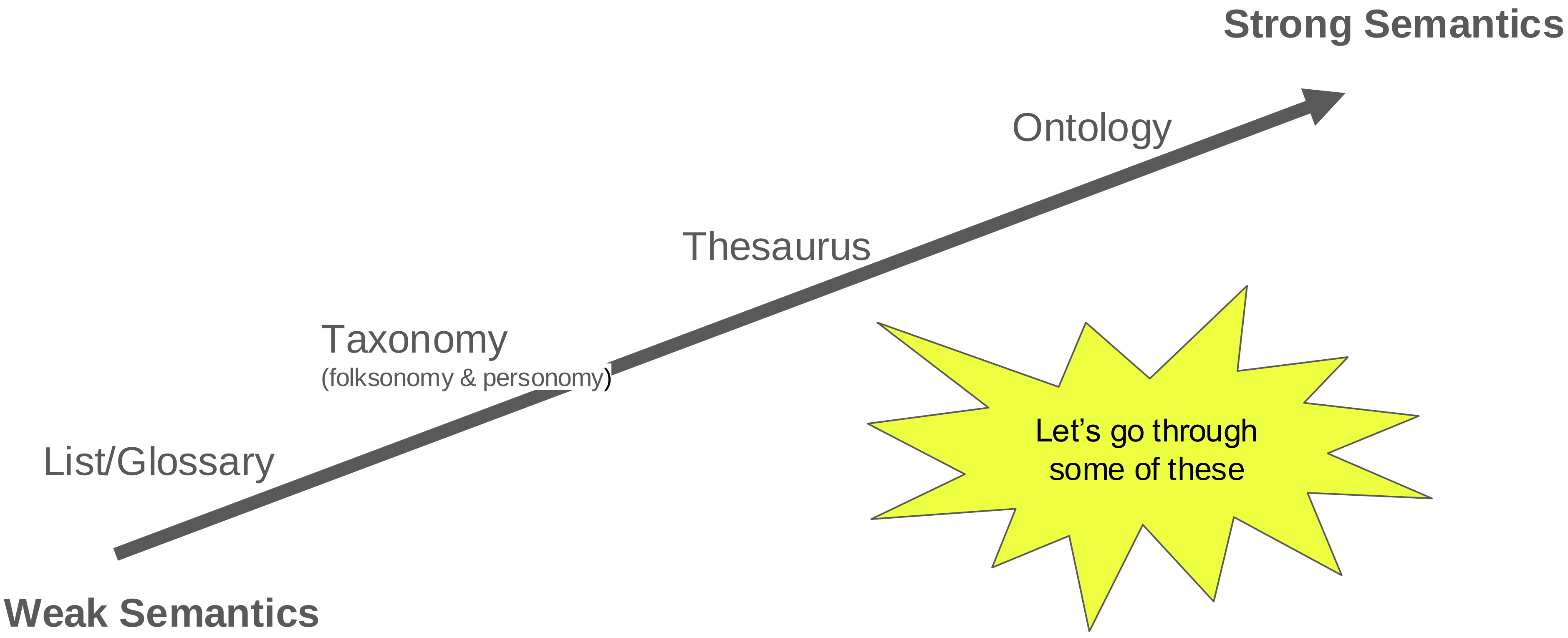
Taxonomy
(folksonomy & personomy)

List/Glossary

Weak Semantics

We can **rank** SA’s in terms of how complex their organisation is, in terms of the ‘strength’ of their semantics

Kinds of Semantic Artifacts



Taxonomies, Folksonomies, Personomies and Thesauri

- **Taxonomy**: tends to refer to the **hierarchical arrangements** of concepts in a domain, usually on the basis of **hypernym/hyponym (X is a Y) relations**
  - The classic example here are biological taxonomies such as those of **Linneaus**
- **Folksonomy**: result from asking users to annotate/categorise/tag the elements in datasets usually consist of online content such as photos, forum threads, blog posts.
  - Usually much more **bottom-up** than taxonomies, as consequence they tend to be **messier** and to lack the rigour of classification schemes conceived as top down.

Taxonomies, Folksonomies, Personomies and Thesauri

- **Personomy**: A folksonomy that's restricted to the annotations made by a single user.
- **Thesauri**: Are like taxonomies but they also include equivalence relationships, hierarchical relationships and associative relationships
  - Most well known such (paper based) resource is **Roget's Thesaurus**
  - **WordNets** can also be seen as thesauri

Useful Semantic Web Taxonomies and Thesauri

- **Taxonomy of Digital Research Activities in the Humanities (TaDiRAH)** provides concepts for DH research activities, and is intended to assist community-driven sites and projects organise DH content
- The **European Language Social Science Thesaurus (ELSST)** is a broad-based, multilingual thesaurus for the social sciences.
- The **Getty Archive Art & Architecture Thesaurus® Online**, used for the description of art items, architecture, and material culture
- We will look at **SKOS**, a model for representing such resources (a semantic artifact for representing semantic artifacts(!)) in later slides

Focus on: Princeton WordNet

- “WordNet® is a large **lexical database** of English. **Nouns**, **verbs**, **adjectives** and **adverbs** are grouped into **sets of cognitive synonyms (synsets), each expressing a distinct concept**. Synsets are interlinked by means of conceptual-semantic and lexical relations.

- WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, **WordNet interlinks not just word forms—strings of letters—but specific senses of words.** As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, **WordNet labels the semantic relations among words**, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity.”

Source: https://wordnet.princeton.edu/

Synsets

- The main relation among words in WordNet is **synonymy**. Synonyms are defined as **words that denote the same concept and are interchangeable in many contexts** and are grouped into unordered sets (synsets). Each of WordNet's 117 000 synsets is linked to other synsets by means of "conceptual relations."
- A synset contains a brief definition ("gloss") and, in most cases, one or more short sentences illustrating the use of the synset members. Word forms with several distinct meanings are represented in as many distinct synsets. Thus, each form-meaning pair in WordNet is unique.
- The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation), but we can also have:
  - **Meronymy**
  - **Antonymy** (for adjectives)
  - **Cross-PoS relations**, e.g. morpho-semantic links

Source: https://wordnet.princeton.edu/. See also: Fellbaum, Christiane (2005). WordNet and wordnets. In: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670.

Beyond English: the Open Multilingual WordNet project

- It aims at interlinking WordNets in multiple languages, creating a shared format. There are currently two ways of connecting WordNets together: via the Princeton WordNet or using the Collaborative Interlingual Index.

- There are two approaches for creating a new WordNet: the expand method and the merge method: **the expand method starts from the structure of the Princeton WordNet** and translates the synsets into the target language, creating additional synsets for specific senses. **The merge method creates the WordNet from scratch and then adds links it to the Princeton WordNet** with cross-lingual links

Source: https://omwn.org/index.html

WordNets in Linked (Open) Data

Cimiano, Chiarcos, McCrae, Gracia 2020, *Linguistic Linked Data*
Alla p. 215: Applying Linked Data Principles to Linking Multilingual Wordnets

- Wordnets are the most widely used lexical resources in natural language processing and computational lexicography
- LOD principles have been applied to the development of the Global WordNet Grid (GWG) in order to form a catalogue of interlingual contexts that extends beyond the Princeton WordNet. Linked Data technologies are used to create a Collaborative Interlingual Index (CILI) that builds on standard Linked Data vocabularies and the RDF data model

What are Ontologies?

- Ontologies are absolutely central to the Semantic Web, but what are they?
- One of the most popular definitions of the term* goes as follows:
    - “**An ontology is a formal, explicit specification of a shared conceptualization**” (Studer et. al.)
- The **‘formal’** part refers to the fact that ontologies are usually written in a **formal language** with a mathematically defined syntax and semantics
- The ‘**shared conceptualisation**’ part emphasises the fact that ontologies deal with knowledge that is **shared** by a community but that is often **implicit** and/or **taken for granted**.
- Ontologies make this implicit knowledge more accessible (explicit)

*We are of course using the term ontology in the informatics and not philosophical sense here

What are Ontologies?

- More specifically ontologies allow us to model the **classes** and **individuals** of interest in a domain, the properties that pertain to them and the **relationships** that hold between them.
- They enable the description of more **complex concepts and relationships** in terms of **simpler concepts and relationships**. In this way we can represent their meanings.
- Ontologies are often written in languages such as OWL (which we'll look at later), for which there exist **semantic reasoners/inference engines** which allow us to automatically check constraints have been adhere to in our data as well as to derive new knowledge from them.

What are Ontologies Used for?

- Ontologies have many different uses. One of the main ways that they're used is to define terms in a **controlled vocabulary** in a machine actionable way. These are lists of terms belonging to some specific domain that have been organised in a meaningful way and which can be used to index and retrieve texts.
- There are other ways of doing this (for example SKOS which we will look at later), but ontologies allow us a lot of **expressivity and flexibility**
- The concepts in an ontology can be given a unique identifier which means that they can be used to **annotate** lots of different datasets allowing us to **integrate** them and **query** them at the same time
- OWL is the Semantic Web ontology language which allows us to make use of the technologies and standards in the stack, e.g., URIs to identify concepts, as well as building on top of RDF and RDFS

What are Ontologies Used for?

- Ontologies have had the most success in the biomedical domain where they have been used for the following
    - Annotation with standard identifiers, in order to integrate together and **query multiple datasets** (e.g., Gene Ontology)
    - As vocabularies for applications relying on **domain-specific terms** (e.g., text mining using ontology labels)
    - **Reasoning over ontology annotated datasets** (e.g., determining which protein family a protein belongs to)
    - Data Mining and Analysis, **using ontologies as background knowledge** (e.g., Gene Set Enrichment Analysis)
- Many of these tasks can also be carried over to other domains…but in the humanities there tends to be **less agreement** than in the hard sciences on basic definitions on the basis of which we can built ontologies

What are Ontologies Made Out Of?

- Abstracting away from technical particulars, ontologies are used to describe three kinds of entity:
  - **Classes/concepts**, Person, Country, Sheep, Author, Word etc
  - **Properties (or relations)** X child of Y, X lives in Y, X is located in Y, X loves B, S isSenseOf W
  - **Individuals** Fahad Khan, Giulia Pedonese, Michele Mallia, Portugal, Europe, Euro 2034, number 23, the word *saudade*
- This is echoed in the division of ontologies into a **TBox** (**T**erminological Box) and an **ABox** (**A**ssertion Box) and sometimes a separate **RBox** (**R**ole/relation Box) too.
- These 3 kinds of entity become the primitive components out of which formal ontology languages such as OWL are constructed.

Excursus: Differences between Ontologies/Taxonomies

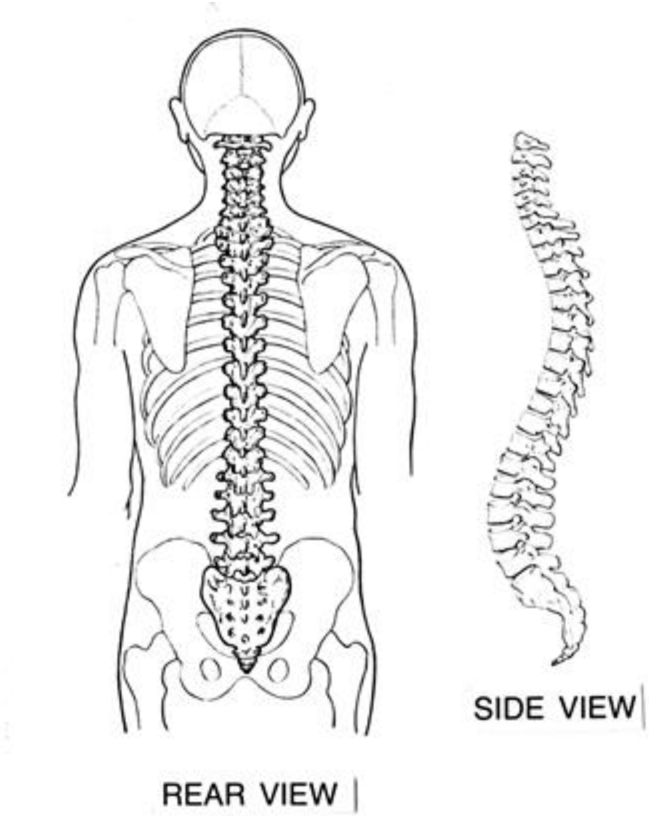
- In practice the term **taxonomy** is often used to refer to ontologies which use only the **subsumption** and (often) **equivalence** relations to organise concepts.
- The subsumption relation in taxonomies and ontologies ($\sqsubseteq$) corresponds to **hyponymy** in lexical semantics, an area where we also describe taxonomies
- However as argued in (Cruse 86) a **well formed** (lexical) taxonomical hierarchy shouldn't include statements like Ewe $\sqsubseteq$ Sheep but only those like Sheep $\sqsubseteq$ Animal and Horse $\sqsubseteq$ Animal. The former division isn't like the latter two.
- Well formed taxonomies should classify things into kinds; therefore we should use restricted criteria such as for instance:
  - X $\sqsubseteq$ Y in a taxonomy if an X is a **kind/type of** Y

Excursus: Differences between Ontologies/Taxonomies

- So we can say:
  - A spaniel is **a kind of** dog
  - A rose is **a type of** flower
  - A mango is **a kind of** fruit
- But the following seem less acceptable:
  - ?A kitten is **a type o**f cat
  - ?A queen is **a type of** monarch
  - ?A waiter is **a kind o**f woman
- Cruse defines this new restricted version of the hyponymy relation **taxonomy** (not to be confused with taxonomy as a kind of resource, semantic artifact).
- His reasoning seems to apply not just to taxonomies in lexical semantics
- Can help us to clear up how to structure (well) Semantic Web taxonomies

Excursus: Differences between Ontologies/Taxonomies

- Interestingly the well known ontology design methodology **Ontoclean** makes very similar points (though using a much more sophisticated analysis) and views the ontological subsumption (IS A) relation **similarly to Cruse's taxonomy relation**.
- In fact we can see a well-defined taxonomy as **the backbone** of an well-defined ontology.
- This is one way of understand how taxonomies and ontologies can be related together



Top-Level Ontologies

- A lot of the vocabularies, we will look at for creating our own linked data language resources should be viewed as **ontologies** -- or at least semantic artifacts. One very useful class of ontologies are **Top-Level Ontologies**, also known as **Foundational Ontologies** or **Upper Ontologies**
- These are ontologies which describe the most general concepts which might be referred to in any individual ontology, e.g.,., **Event, Individual, Agent,...**
- Usually these are concepts which **aren’t specific to any one domain** and can be re-used by other ontologies in different domains;
    - Although in some cases they tend to be more associated with particular domains or sets of domains, e.g., CIDOC-CRM for the humanities
- Using a Top-level Ontology we affirm the most **fundamental ontological commitments** made by a domain specific ontology.

Top-Level ~~Ontologies~~ Semantic Artifacts

- Means we don't need to **reinvent the wheel**. Also help to ensure **a basic level of interoperability** among other more specific ontologies based on the same top level ontology, because of explicitly shared ontological commitments
- Top-level ontologies can often just be a series of **useful classes and properties** (in fact many would be better called **top level semantic artifacts**) or they can be much more extensive and incorporate sophisticated metaphysical reasoning (e.g., **DOLCE** and **BFO**)
- We can also have ontologies which describe general concepts in a domain; these are **Mid Level Ontologies**, mid way between a top level ontology and a more specific domain ontology

Some Useful Top Level SAs

- Well known Top-Level Ontologies (Semantic Artifacts) include:
  - **Dublin Core**
  - **Friend of a Friend (FOAF)**
  - **Schema.org**
  - **DOLCE**
  - **BFO**
  - **SUMO**
- These are listed in ***A Survey of Top Level Ontologies*** by the UK's **Digital Twin Initiative:**
  - https://www.cdbb.cam.ac.uk/files/a_survey_of_top-level_ontologies_lowres.pdf

Some Useful Top Level SAs

- We will look at the first three of these top level ontologies in the following slides, i.e., **Dublin Core**, **FOAF**, **Schema.org**
- We will also look at **DCAT** & **CIDOC-CRM** the latter of which is regularly used in the Digital Humanities
- These ontologies will likely be useful if you are defining your own ontology (or in fact for many other Semantic Web related use cases!)
- They are also important for creating **metadata** for e.g., **language resources.**
- Let’s start with the Dublin Core!

Dublin Core

- A set of vocabulary terms used to describe **web resources** (such as video, images, web pages), and physical resources such as books or artworks.
- Developed in 1995 during a workshop in **Dublin, Ohio**.
- Intended to facilitate the **discovery of electronic resources** by providing a **simple** and standardised set of conventions for resource description.
- Helped standardise metadata descriptions on the Semantic web
- Used commonly for metadata in all categories of Semantic Web resources, including for LLOD cloud resources

Dublin Core Categories

or: “An entity responsible for making contributions to the resource.”

e: “The spatial or temporal topic of the resource, the spatial applicability
ource, or the jurisdiction under which the resource is relevant.”
An entity primarily responsible for making the resource.”
oint or period of time associated with an event in the lifecycle of the

on: “An account of the resource.”

Dublin Core Categories Ctd

- **Format:** “The file format, physical medium, or dimensions of the resource.”
- **Identifier:** “An unambiguous reference to the resource within a given context.”
- **Language:** “A language of the resource.”
- **Publisher:** “An entity responsible for making the resource available.”
- **Relation:** “A related resource.”

Dublin Core Categories Ctd

- **Rights:** "Information about rights held in and over the resource."
- **Source:** "A related resource from which the described resource is derived."
- **Subject:** "The topic of the resource."
- **Title:** "A name given to the resource."
- **Type:** "The nature or genre of the resource."

Dublin Core

- abstract
- accessRights
- accrualMethod
- accrualPeriodicity
- accrualPolicy
- alternative
- audience
- available
- bibliographicCitation
- conformsTo
- *contributor*
- *coverage*
- created
- *creator*
- *date*
- dateAccepted
- dateCopyrighted
- dateSubmitted
- *description*
- educationLevel
- extent
- *format*
- hasFormat
- hasPart
- hasVersion
- *identifier*
- instructionalMethod
- isFormatOf
- isPartOf
- isReferencedBy
- isReplacedBy
- isRequiredBy
- issued
- isVersionOf
- *language*
- license
- mediator
- medium
- modified
- provenance
- *publisher*
- references
- *relation*
- replaces
- requires
- *rights*
- rightsHolder
- *source*
- spatial
- *subject*
- tableOfContents
- temporal
- *title*
- *type*
- valid

Dublin Core - Your Turn

- Create a DC dataset using the DC Template
  - http://metadataetc.org/dctemplate.html

Friend of a Friend (FOAF)

- An ontology describing **persons, their activities, and their relations to other people and objects.**
- Origin: Developed by Dan Brickley and Libby Miller in **2000**.
- Enables the creation of a web of machine-readable pages describing people, groups, the links between them, and the things they create and do.
- Useful in describing authors, contributors, etc, to a resource

Friend of a Friend (FOAF)

FOAF Core

-
  - Agent
  - Person
  - name
  - title
  - img
  - depiction (depicts)
  - familyName
  - givenName
  - knows
  - based_near
  - age
  - made (maker)
  - primaryTopic (primaryTopicOf)
-
  - Project
  - Organization
  - Group
  - member
-
  - Document
  - Image

Social Web

- nick
- mbox
- homepage
- weblog
- openid
- jabberID
- mbox_sha1sum
- interest
- topic_interest
- topic (page)
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- publications
- currentProject
- pastProject
- account
- OnlineAccount
- accountName
- accountServiceHomepage
- PersonalProfileDocument
- tipjar
- sha1
- thumbnail
- logo

Friend of a Friend (FOAF)

Create a DC dataset using FOAF-a-matic:

- http://ldodds.com/foaf/foaf-a-matic.html

Schema.org

- **Schema.org** is a **collaborative**, **community** activity with a mission to create, maintain, and promote **schemas for structured data on the Internet**, on web pages, in email messages, and beyond
- Proposed as a selection of schemas **for adding metadata annotations** to HTML pages.
- Recently started to become a very popular ontology.

Schema.org


Place
Accommodation
Apartment
CampingPitch
House
SingleFamilyResidence
Room
HotelRoom
MeetingRoom
Suite
AdministrativeArea
CivicStructure
Landform
LandmarksOrHistoricalBuildings
LocalBusiness
Residence
ApartmentComplex
GatedResidenceCommunity
TouristAttraction
TouristDestination

DCAT

- **Data Catalog Vocabulary (DCAT)** developed by the World Wide Web Consortium (W3C), is a vocabulary & designed to facilitate interoperability between **data catalogs published on the web,** providing a way to describe datasets in catalogs to make them more discoverable and accessible.
- Key components of DCAT include:
  - **Dataset**: Represents a collection of data, which could be in various formats.
  - **Distribution**: Represents an accessible form of a dataset, such as a downloadable file.
  - **Catalog**: Represents a collection of datasets.

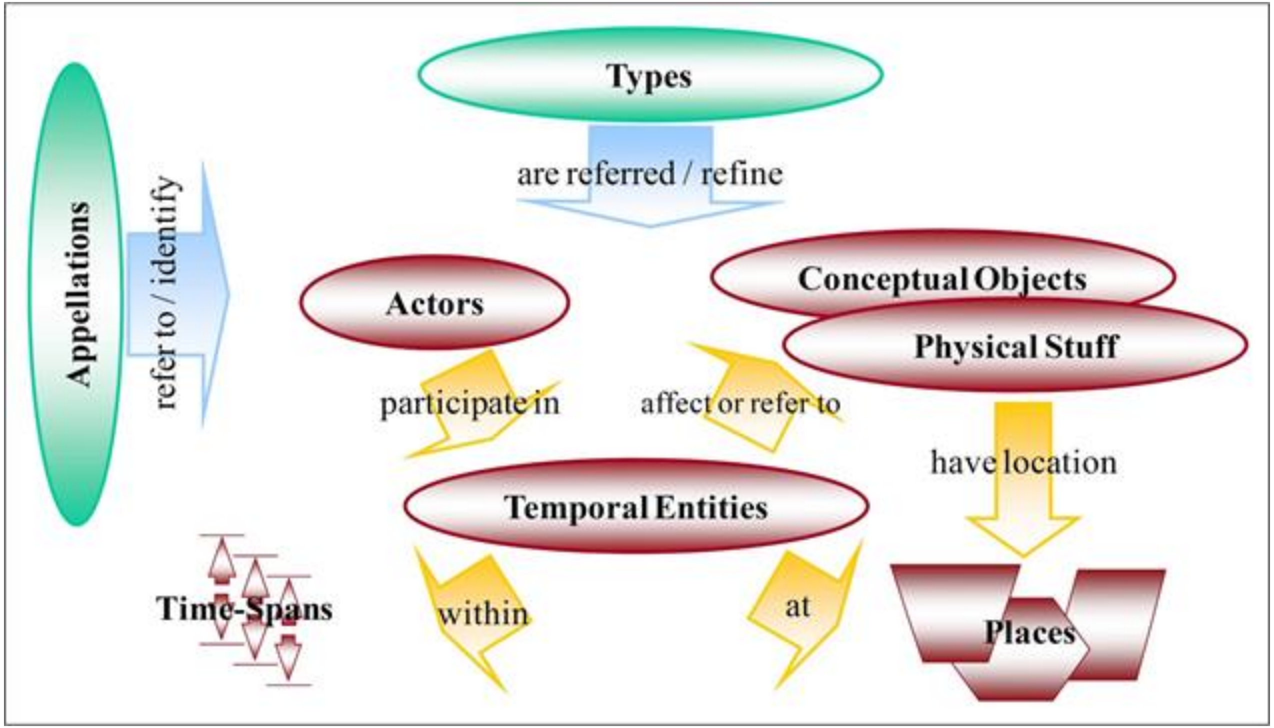
CIDOC CRM

- We have looked at generic top level ontologies/SAs, let's look at those associated with different domains
- **CIDOC CRM** is a Top-Level Ontology (CRM= Conceptual Reference Model) originally created for the cultural heritage domain
- Published as an **ISO Standard** in 2006
- Intended to mediate between **cultural heritage datasets** in order to enable information exchange and data integration
- CIDOC CRM has proven itself an important tool in establishing **interoperability between individual resources** through making descriptions of objects semantically transparent via ontological concepts and properties, that is by rendering implicit knowledge explicit using CIDOC concepts

CIDOC CRM

- The CIDOC CRM ontology consists of a number of real world concepts linked together using properties to which data is aligned
- CIDOC CRM is very much **event-centric** and organised in terms of **Things**, that persist through time, and **Events** which ‘happen’ rather than ‘are’

CIDOC CRM



From https://isl.ics.forth.gr/mapping_technology/page/cidoc-crm

CIDOC CRM

Object type: tanto; short sword-sheath; menuki; kozuka; hilt; fuchi-kashira; blade
Museum number: 1992,0523.2
Description: Sword blade (tanto); with mounting (short sword-sheath; kozuka; hilt; menuki; fuchi-kashira). Blade: made of steel; signed. Sheath: made of black lacquered wood. Hilt: with gold mekugi; made of wood and skin (ray). Kozuka: crane in high-relief coloured metal inlay on silver ground; inscribed. Menuki: in shape of corn?; made of gilded metal. Fuchi-kashira: made of black lacquered metal. Soshu school blade and Goto school metal fittings.
Producer name: Made by: Goto Ichijo (metal fittings); Made by: Shintogo Kunimitsu (blade)
Culture/period: Meiji Era (metal fittings); Kamakura Period (blade)
Date: 14thC (early; blade); 19thC (late; metal fittings)
Production place: Made in: Japan (Asia,Japan)
Materials: wood; steel; silver; ray skin; metal; lacquer; gold
Technique: lacquered; inlaid ; high relief; gilded; colour
Inscriptions:
Inscription Type: signature
Inscription Script: Japanese
Inscription Position: blade, tang, obverse
Inscription Content: 国光; Inscription Transliteration; Kunimitsu, etc
Curator's comments: Harris 2005 - 'Hira zukuri' tanto blade with the slight 'uchizori' curve of the late Kamakura period. The blade has 'itame' with 'mokume' grain with 'jifu utsuri' and much 'chikei'. The 'suguha hamon' is of fine 'nie' with 'kinsuji'. The maker, Shintogo Kunimitsu, is feted as the founder of the Soshu tradition at Kamakura in the late Kamakura period.
Bibliography: Harris 2005 fig. 11, col. pl. 11, 12 bibliographic details
Location: G93/case10
Exhibition history
Exhibited: 2006 Oct 13-, BM Japanese Galleries, 'Japan from prehistory to the present'
Subjects: arms/armour term details;
Acquisition name: Purchased through: Eskenazi Ltd biography; Purchased from: Christie's biography; Previous owner/ex-collection: Dr Walter A Compton biography
Acquisition date: 1992
Acquisition notes: Bought at Christie's (lot 226) by Eskanazi Ltd at the BM's request. Former collection of Walter A Compton.
Department: Asia
Registration number: 1992,0523.2

CIDOC CRM

Object type: tanto; short sword-sheath; menuki; kozuka; hilt; fuchi-kashira; blade
Museum number: 1992,0523.2
Description: Sword blade (tanto); with mounting (short sword-sheath; kozuka; hilt; menuki; fuchi-kashira). Blade: made of steel; signed. Sheath: made of black lacquered wood. Hilt: with gold mekugi; made of wood and skin (ray). Kozuka: crane in high-relief coloured metal inlay on silver ground; inscribed. Menuki: in shape of corn?; made of gilded metal. Fuchi-kashira: made of black lacquered metal. Soshu school blade and Goto school metal fittings.
Producer name: Made by: Goto Ichijo (metal fittings); Made by: Shintogo Kunimitsu (blade)
Culture/period: Meiji Era (metal fittings); Kamakura Period (blade)
Date: 14thC (early; blade); 19thC (late; metal fittings)
Production place: Made in: Japan (Asia,Japan)
Materials: wood; steel; silver; ray skin; metal; lacquer; gold
Technique: lacquered; inlaid ; high relief; gilded; colour
Inscriptions:
Inscription Type: signature
Inscription Script: Japanese
Inscription Position: blade, tang, obverse
Inscription Content: 国光; Inscription Transliteration; Kunimitsu, etc
Curator's comments: Harris 2005 - 'Hira zukuri' tanto blade with the slight 'uchizori' curve of the late Kamakura period. The blade has 'itame' with 'mokume' grain with 'jifu utsuri' and much 'chikei'. The 'suguha hamon' is of fine 'nie' with 'kinsuji'. The maker, Shintogo Kunimitsu, is feted as the founder of the Soshu tradition at Kamakura in the late Kamakura period.
Bibliography: Harris 2005 fig. 11, col. pl. 11, 12 bibliographic details
Location: G93/case10
Exhibition history
Exhibited: 2006 Oct 13-, BM Japanese Galleries, 'Japan from prehistory to the present'
Subjects: arms/armour term details;
Acquisition name: Purchased through: Eskenazi Ltd biography; Purchased from: Christie's biography; Previous owner/ex-collection: Dr Walter A Compton biography
Acquisition date: 1992
Acquisition notes: Bought at Christie's (lot 226) by Eskanazi Ltd at the BM's request. Former collection of Walter A Compton.
Department: Asia
Registration number: 1992,0523.2

CIDOC CRM

- Used in a number of very successful projects and initiatives, to render the information coming from various catelogues interopeable, including Map the Manuscript Migrations
- CIDOC CRM has several other modules that build on the success of the original model including **a provenance model,** a **spatiotemporal model**, a model for archeology and buildings
- FRBRoo (Version 0.2 released in 2013) an attempt to create an ontology that harmonises both CIDOC CRM and the Functional Requirements for Bibliographic Records (FRBR), data model; this changed name to become LRM.
  - Khan and Salgado propose the integration of OntoLex with FRBRoo (LRM) to model lexicons and dictionaries as both physical, visual and informational objects.

VIAF, DBpedia, and Wikidata

- Now, we will focus on three important general purpose Semantic Web ontological resources which should come up useful in all kinds of tasks:
  - **VIAF**
  - **DBpedia**
  - **Wikidata**
- These resources are useful for adding references in dictionary citations to proper names (geographical locations, names of authors, works)
- Shows the potential for representing extra-linguistic information in language resources in a standardised and highly machine actionable way (something which the Semantic Web facilitates)

VIAF

- **Virtual International Authority File**
- Authority File: Assigns **a unique identifier** to people, topics, organisations, etc
- VIAF offers **URIs for use in Linked Data datasets**
- Luís de Camões
  - https://viaf.org/viaf/186297237/
- Universidade nova de Lisboa:
  - https://viaf.org/viaf/124928771/

DBpedia and Wikidata

- Both **DBpedia** and **Wikidata** are based on Wikipedia
- Differences:
    - DBpedia: An RDF ontology that is generated from Wikipedia **infoboxes**
    - Wikidata: An ontology created to **enrich** Wikipedia

Wikidata

- Can be edited **directly** by users
- Consists of items with an **Q-prefix id** and properties with a **P-prefix id**
- Every item is the subject of a number of claims/triples
  - These claims can be qualified with qualifiers and annotated with references
- Wikidata page for **Luís de Camões**:
  - https://www.wikidata.org/wiki/Q590
- Wikidata page for **Taiwan**:
  - https://www.wikidata.org/wiki/Q865

Wikidata

- Wikidata is **becoming popular in the humanities**
- Used as a data source as well as for **storing data**
- Map of residences of accused witches based on wikidata:
    - https://witches.is.ed.ac.uk/
- Use of Wikidata to document a local music scene:
    - https://www.bpl.org/blogs/post/boston-rock-city-explore-wikidata-and-learn-about-local-music/
- We will look at the lexicographic data hosted in Wikidata in tomorrow’s lesson

Useful Semantic Artifacts on the LLOD Cloud

Lexvo

- A linked data SA that provides information about languages, words, and other linguistic entities
- Each language in lexvo is assigned a unique **URI**, which can be used to unambiguously refer to that language.
  - For example, the URI for the English language is `http://lexvo.org/id/iso639-3/eng`.
- **Lexical Entries**: Lexvo includes lexical entries that provide information about words and phrases in various languages. These entries can include **definitions**, **translations**, and **phonetic transcriptions**.
- **Language Relationships**:The service provides information about the relationships between languages, such as which languages are dialects of others or belong to the same language family.
- **Cultural and Geographic Data**: Lexvo links languages **to the regions** where they are spoken and to the cultures that use them, providing **a contextual understanding** of each language

Useful Semantic Artifacts on the LLOD Cloud

Lexinfo:

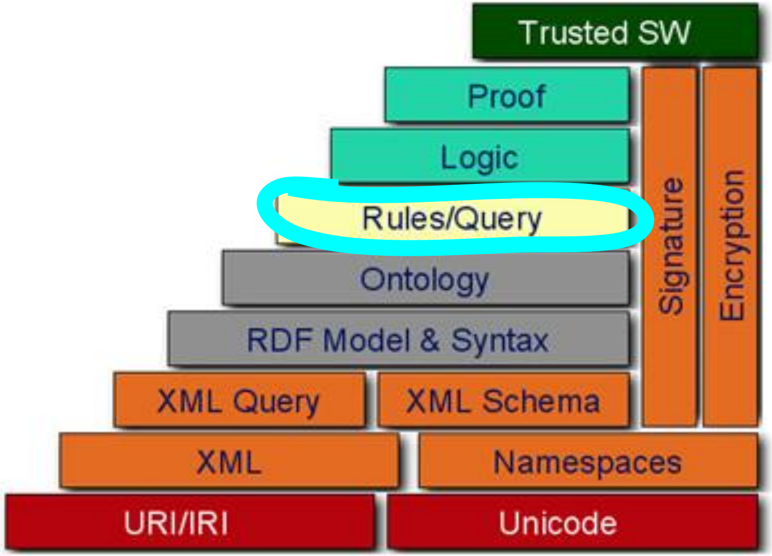
- “[A]n ontology that was defined during the Monnet Project to provide **data categories** for the Lemon model [predecessor]. It has since since been updated with the new OntoLex-Lemon model of the OntoLex community group.”
- It provides e.g., **part of speech** values, other relevant **morpho-grammatical** properties, for OntoLex lexicons, as we will see….
- Based on the **Old ISOCat registry**, since discontinued
- Difficult to navigate, a real **hodgepodge**
- Shows that ontologies aren’t always well thought out (even if they are useful)

Search Engines/Repositories

- Linked Open Vocabularies:
  - https://lov.linkeddata.es/dataset/lov
- Another RDF vocabulary search engine
  - http://vocab.cc/
- Specialised ontology repositories:
  - Archivio: http://vocab.cc/

Exploring the Semantic Web Stack

A second tutorial on SPARQL (with Wikidata)



Returning to SPARQL

- We return to **SPARQL** this time using the **Wikidata** SPARQL endpoint
- Recall in Wikidata (differently from DBpedia) the IDs are **arbitary** (and **not human readable**), e.g., https://www.wikidata.org/wiki/Q311145, https://www.wikidata.org/wiki/Property:P800
- With the **Q prefix** used for items and the **P prefix** for properties
- We will look at a few queries from the **Wikidata Query point** https://query.wikidata.org/
- We take these from the Wikidata tutorial https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

First Wikidata Query

Who were the sons of **Johann Sebastian Bach (Q1339),** where the relevant property `(father of)` has the **ID P:P22**:

```
SELECT ?child

WHERE

{

# ?child  father   Bach

  ?child wdt:P22 wd:Q1339.

}
```

Material from https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

SERVICE

- The **SERVICE** keyword in SPARQL is used to query **a remote SPARQL endpoint** from within a SPARQL query. This allows **federated querying**, where data from **different SPARQL endpoints** can be combined in a single query.
- In Wikidata the **SERVICE keyword** can be used to access a specific service provided by the Wikidata SPARQL endpoint to **fetch labels** for the queried entities in the desired language.

Updated Wikidata Query

```
SELECT ?child ?childLabel

WHERE

{

# ?child  father   Bach

  ?child wdt:P22 wd:Q1339.

  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]". }

}
```

- **wikibase:label** is a predefined service in Wikidata that helps in fetching the human-readable labels (names) for the entities.
- **bd:serviceParam wikibase:language "[AUTO_LANGUAGE]"** specifies the language parameter. [AUTO_LANGUAGE] is a placeholder that automatically selects the language based on the user's preferences or the default language of the endpoint.

Material from https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

Property Paths

Property paths in SPARQL are a powerful feature that **allows querying of complex relationships** between RDF resources by specifying patterns of predicates (properties) in the query. Property paths can be used to **navigate through RDF graphs**, allowing more flexible and expressive queries.

A simple path specifies a direct relationship between two nodes. However the sequence path/allows **chaining** multiple predicates,

E.g., `?item wdt:P31/wdt:P279/wdt:P279 ?class.`

The **zero-or-more path *** matches zero or more occurrences of a predicate.

E.g., `?item wdt:P31/wdt:P279* ?class.`

The **one-or-more path +** matches one or more occurrences of a predicate.

E.g., `?item wdt:P31/wdt:P279+ ?class.`

Material from https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

Exercise

How can I modify the following query to find all the descendants of Bach where P40 is the child relation?

```
SELECT ?d ?dLabel

WHERE

{

  wd:Q1339 wdt:P40 ?d.

  SERVICE wikibase:label { bd:serviceParam wikibase:language
"[AUTO_LANGUAGE]". }

}
```

Material from https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

Exercise

Answer

```
SELECT ?d ?dLabel
WHERE
{
  wd:Q1339 wdt:P40+ ?d.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]". }
}
```

Material from https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

An Alternative

```
SELECT ?descendant ?descendantLabel
WHERE
{
  ?descendant (wdt:P22|wdt:P25)+ wd:Q1339.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]". }
}
```

Material from https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

BIND, BOUND, IF

Recall: We can use **BIND(expression AS ?variable)** to assign the result of an expression to a variable; **BOUND(?variable)** to test if a variable has been bound to a value (useful for variables introduced in an **OPTIONAL** clause); we can additionally add conditionals to SPARQL queries via **IF(condition,thenExpression,elseExpression)**

Query

We would like to write a query that shows the name of all **French writers born in the second half of the 18th century**, but instead of just showing their label, it shows their **pseudonym (P742)** if they have one, and **only the label** if a pseudonym doesn't exist.

First Part

French writers from the second half of the 18th century

```
SELECT ?writer
WHERE
{
   ?writer wdt:P31 wd:Q5;  # An instance of (P31) human (Q5)
                wdt:P27 wd:Q142;  # a citizen (P27) of France (Q142)
           wdt:P106 wd:Q36180;  # with occupation (106) writer (Q36180)
          wdt:P569 ?dob.   # and dob (P569)
  FILTER("1751-01-01"^^xsd:dateTime <= ?dob && ?dob < "1801-01-01"^^xsd:dateTime).
}
```

Second Part

French writers from the second half of the 18th century with their English label, get the pseudonym if it exists

```
SELECT ?writer ?writerLabel WHERE
{  ?writer wdt:P31 wd:Q5;
          wdt:P27 wd:Q142;
          wdt:P106 wd:Q36180;
          wdt:P569 ?dob.
  FILTER("1751-01-01"^^xsd:dateTime <= ?dob && ?dob < "1801-01-01"^^xsd:dateTime).
  # get the English label
  ?writer rdfs:label ?writerLabel.
  FILTER(LANG(?writerLabel) = "en").
 # get the pseudonym, if it exists
  OPTIONAL { ?writer wdt:P742 ?pseudonym. }
}
```

QUERY

Full query

```
SELECT ?writer ?label

WHERE

{....

   ?writer rdfs:label ?writerLabel.

        FILTER(LANG(?writerLabel) = "en").

       # get the pseudonym, if it exists

  OPTIONAL { ?writer wdt:P742 ?pseudonym. }
BIND(IF(BOUND(?pseudonym),?pseudonym,?writerLabel) AS ?label).

}
```

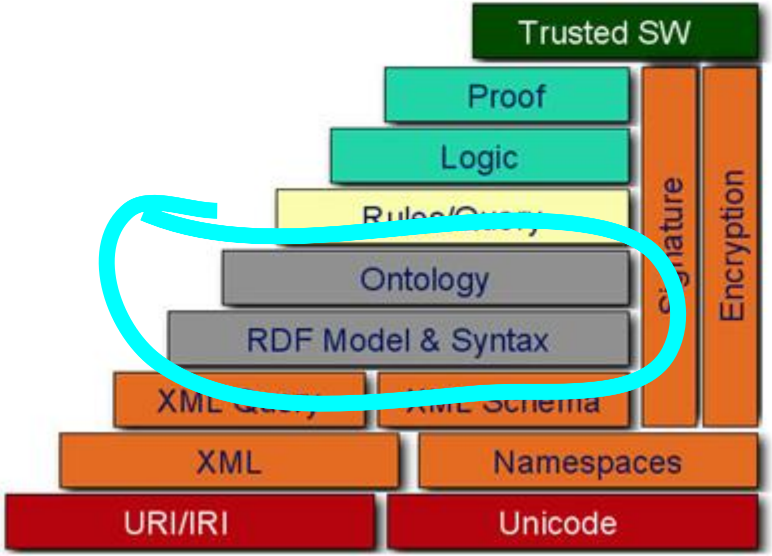
QUERY

Full query

```
SELECT ?writer ?label

WHERE

{....

   ?writer rdfs:label ?writerLabel.

       FILTER(LANG(?writerLabel) = "en").

      # get the pseudonym, if it exists

  OPTIONAL { ?writer wdt:P742 ?pseudonym. }
BIND(IF(BOUND(?pseudonym),?pseudonym,?writerLabel) AS ?label).

}
```

This conditional choose between the two options

Exploring the Semantic Web Stack



SKOS

- We have looked at Semantic Artifacts in theory but how can we actually build them ourselves, what languages should we use (in addition to RDF(S)?
- SKOS (Simple Knowledge Organisation System) is a Semantic Web model for building RDF taxonomies/thesaurui; it is a nice, lightweight alternative to OWL in cases where we don't need to create an ontology, but is more flexible than RDFS
- SKOS is concept-centric model in which terms are attached as labels to concepts, and concepts can be arranged in hierarchies using the broader and narrower properties, or associated together using the related property
- There exist many tools for editing or managing SKOS datasets, including several of those which are also used for working with OWL ontologies

SKOS

- SKOS is based around the core SKOS class Concept
    - e.g., ex:animals rdf:type skos:Concept
- SKOS concepts can have labels in different languages, these labels can be either preferred labels or alternative labels
    - e.g., ex:animals rdf:type skos:Concept;

  skos:prefLabel "animals"@en;

  skos:altLabel "creatures"@en;

  skos:prefLabel "animaux"@fr;

  skos:prefLabel "animali"@it.

SKOS

```
ex:animals rdf:type skos:Concept;
        skos:prefLabel "animals"@en;
      skos:narrower ex:vertebrates.
ex:vertebrates rdf:type skos:Concept;
        skos:prefLabel "vertebrates"@en;
        skos:broader ex:animals;
      skos:narrower ex:mammals.
ex:mammals rdf:type skos:Concept;
      skos:prefLabel "mammals"@en;
      skos:broader ex:vertebrates.
```

SKOS-XL

Sometimes it is required to describe relationships among lexical labels representing concepts. For this SKOS provides an extension SKOS-XL - SKOS eXtension for Labels - to identify, describe and link lexical labels.

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://www.example.com/> .
@prefix skosxl: <http://www.w3.org/2008/05/skos-xl#> .

ex:europeanWildcat rdf:type skos:Concept;
  skosxl:prefLabel ex:europeanWildcatLabel1;
  skosxl:altLabel ex:europeanWildcatLabel2.

ex:europeanWildcatLabel1 rdf:type skosxl:Label;
  skosxl:literalForm "European wildcat"@en.

ex:europeanWildcatLabel2 rdf:type skosxl:Label;
  skosxl:literalForm "Felis silvestris"@la.

ex:europeanWildcatLabel1 skosxl:labelRelation ex:europeanWildcatLabel2.
```

Ksenia Zaytseva and Matej Ďurčo (2020). Controlled Vocabularies and SKOS. Version 1.1.0. Edited by Matej Ďurčo and Tanja Wissik. DARIAH-Campus. [Training module]. https://campus.dariah.eu/id/D8d6OrLdpLlGRqBSQDVN0

Controlled vocabularies

- CV are knowledge organization systems that contain (optionally) structured set of concepts/terms for organizing and classifying data in order to ensure its future access and retrieval.
  - The concepts/terms are data descriptors **related to each other via explicit relationships** (**hierarchical** or **associative**).
  - These data descriptors are used to distinguish and define the characteristics of knowledge resources in a specific domain.
- Using controlled vocabularies the resources **can be queried (!!!)**, retrieved, analysed and linked to other relevant information objects.

Ksenia Zaytseva and Matej Ďurčo (2020). Controlled Vocabularies and SKOS. Version 1.1.0. Edited by Matej Ďurčo and Tanja Wissik. DARIAH-Campus. [Training module]. https://campus.dariah.eu/id/D8d6OrLdpLlGRqBSQDVN0

Types of CV

- **On the Web** → vocabularies are often used in building the information architecture for websites, data repositories, information systems, etc…
- **Biology** → classification of living organisms (e.g. taxonomies of living organisms, classifications of cross-species anatomical entities…).
- **Public health and medicine** → have CVs in various forms (terminologies, thesauri, ontologies) for defining categorizations and classifications for biomedical investigations, diseases, symptoms, medical errors, etc.
- **International organizations** → United Nations terminologies: to standardize terms and translations in international affair to eliminate ambiguity in terms used in international communication.
- **GLAM** (Galleries/Libraries/Archives/Museums) → describe their objects and resources, build catalogues and information systems.
- **Computer Science** → used in data mining, knowledge extraction, or conversation AI use CVs to classify entities and objects in text or speech recognition

Ksenia Zaytseva and Matej Ďurčo (2020). Controlled Vocabularies and SKOS. Version 1.1.0. Edited by Matej Ďurčo and Tanja Wissik. DARIAH-Campus. [Training module]. https://campus.dariah.eu/id/D8d6OrLdpLlGRqBSQDVN0

Vocabulary platform

- A vocabulary platform is an integrated system designed to manage, publish and provide access to controlled vocabularies used in the organisation and classification of knowledge.

- These platforms facilitate the search and navigation of terms through intuitive and multilingual web interfaces, and often include APIs for integration with other applications.

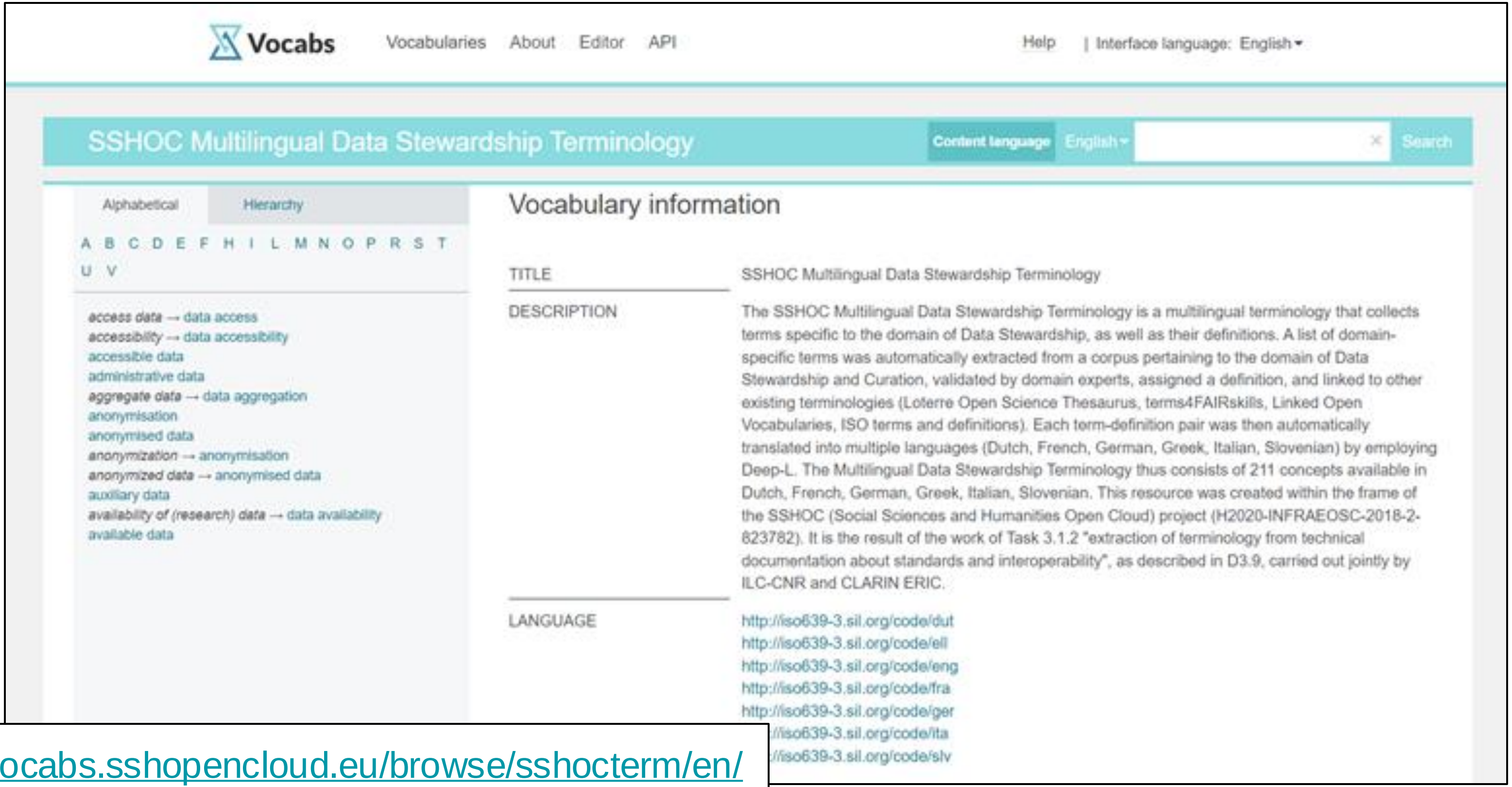
Ksenia Zaytseva and Matej Ďurčo (2020). Controlled Vocabularies and SKOS. Version 1.1.0. Edited by Matej Ďurčo and Tanja Wissik. DARIAH-Campus. [Training module]. https://campus.dariah.eu/id/D8d6OrLdpLlGRqBSQDVN0

SKOSMOS

- Skosmos is an **open source tool** developed for publishing controlled vocabularies represented according to the SKOS (Simple Knowledge Organisation System) model.
- Skosmos offers a **multilingual web interface** that enables the navigation and search of concepts through labels and semantic relations.
- The system architecture is based on a SPARQL endpoint for data storage and uses the Twig template engine to render the user interface.
- The system also supports a **REST API** for integration with other applications, enabling the use of vocabularies in annotation or cataloguing systems.

https://api.finto.fi/doc/

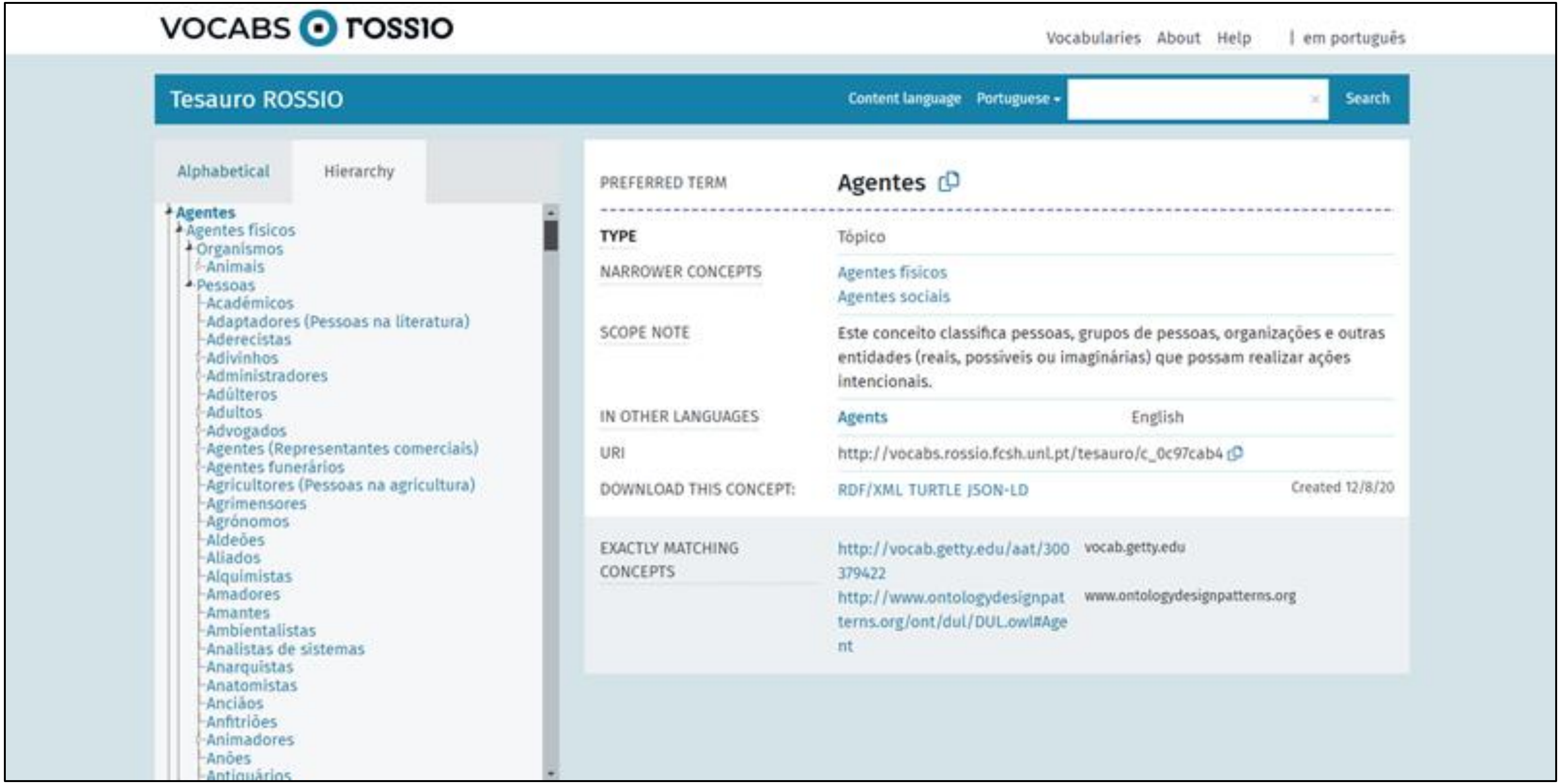
SSHOC Multilingual Data Stewardship Terminology



https://vocabs.sshopencloud.eu/browse/sshocterm/en/

https://vocabs.rossio.fcsh.unl.pt/pub/tesauro/en/page/c_0c97cab4?clang=pt

ROSSIO



Web Ontology Language (OWL)

- We have looked at ontologies in the abstract but how do we actually develop ontologies for the Semantic Web that we can publish and re-use?
- **Answer: We use OWL** a **knowledge representation language** for the semantic web that is built on top of RDF(S)
- ...or rather it's a family of such languages
- The latest version is called **OWL2** and was released as a **W3C recommendation in late 2009**
- Based on previous knowledge representation languages including the DARPA funded DAML

Types of OWL

- OWL comes in three main flavours (in order of increasing expressivity):
    - **OWL Lite**: intended for creating lightweight ontologies, e.g., classification hierarchies, easier to provide tool support
    - **OWL DL**: boasts the maximum expressivity without incurring nasty computational properties such as undecidability. DL stands for Description Logic. Reasoners for OWL DL are slower than those for OWL Lite.
    - **OWL Full**: More expressive but without certain computational guarantees

The Features of OWL

- OWL allows us to **add logical constraints** to the definition of **classes and properties**. The idea is that these constraints can be **automatically checked** by OWL-compatible ontology modelling tools
- OWL also gives us the following very useful properties:
  - `owl:sameAs` (two individuals are the **same**) and `owl:differentFrom` (two individuals are **different**)
    - `dbr:Leonardo_da_Vinci owl:sameAs dbpedia-ja:レオナルド・ダ・ヴィンチ`
    - `dbr:Leonardo_davincii  owl:differentFrom dbr:Leonardo_da_Vinci`
  - `owl:equivalentClass` (two classes are the **same**)

The Features of OWL

- OWL allows us to define our own properties and they can be of two types:
  - **Object properties**: binary relations holding between **instances** of classes
    - `dbr:Lisbon dbo:country dbr:Portugal`
  - **Datatype properties**: binary relations between class **instances** and **RDF literals**
    - `dbr:Lisbon dbp:name “Lisbon”@en`
    - `dbr:Lisbon dbo:populationTotal "544851"^^xsd:nonNegativeInteger`

The Features of OWL

- OWL also allows us to specify the **transitivity** and **symmetry** of properties
  - **Transitivity**: `ancestorOf, i.e., ` if `A ancestorOf B and B ancestorOf C,` then `A ancestorOf C`
  - **Symmetry**: `isSiblingOf`
    - **`dbpedia:Noel_Gallagher ex:isSiblingOf dbpedia:Liam_Gallagher`**
    - **`dbpedia:Liam_Gallagher ex:isSiblingOf dbpedia:Noel_Gallagher`**
- And to constrain roles to be **functional** (if aRb and aRc then b=c) and **inverse functional** (if bRa and cRa then b=c),
  - **Functional**: `populationTotal`
    - **dbr:Lisbon dbp:populationTotal 544851**
- We can also specify that one property is the **inverse of another** (if aRb then $bR^{-}a$)
  - e.g., `hasFather` and `isFatherOf`

The Features of OWL

- OWL also allows us to encode the restrictions **allValuesFrom** and **someValuesFrom**

```
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasPet ;
    owl:allValuesFrom :Animal
] .
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasPet ;
    owl:someValuesFrom :Animal
] .
```

The Features of OWL

- OWL also allows us to encode the restrictions **allValuesFrom** and **someValuesFrom**

```
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasPet ;
    owl:allValuesFrom :Animal
] .
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasPet ;
    owl:someValuesFrom :Animal
] .
```

A person is a kind of thing who can only have an animal as pet

The Features of OWL

- OWL also allows us to encode the restrictions **allValuesFrom** and **someValuesFrom**

```
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasPet ;
    owl:allValuesFrom :Animal
] .
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasPet ;
    owl:someValuesFrom :Animal
] .
```

A person is a kind of thing who has at least one Animal as pet

The Features of OWL

- OWL also allows us to encode the following property restrictions
  - **owl:cardinality**, **owl:maxCardinality**, and **owl:minCardinality**

```
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasParent ;
    owl:cardinality "2"^^xsd:nonNegativeInteger
] .
```

- `:Person rdfs:subClassOf [ rdf:type owl:Restriction ; owl:onProperty :hasParent ; owl:maxCardinality "2"^^xsd:nonNegativeInteger ] .`
- `:Person rdfs:subClassOf [rdf:type owl:Restriction ; owl:onProperty :hasParent ;owl:minCardinality "1"^^xsd:nonNegativeInteger] .`

A person is a kind of thing who has two parents

The Features of OWL

A person is a kind of thing with at most two and at least one parents

- OWL also allows us to encode the following property restrictions
  - **owl:cardinality**, **owl:maxCardinality**, and **owl:minCardinality**

```
:Person rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty :hasParent ;
    owl:cardinality "2"^^xsd:nonNegativeInteger
] .
```

- `:Person rdfs:subClassOf [ rdf:type owl:Restriction ; owl:onProperty :hasParent ; owl:maxCardinality "2"^^xsd:nonNegativeInteger ] .`
- `:Person rdfs:subClassOf [rdf:type owl:Restriction ; owl:onProperty :hasParent ;owl:minCardinality "1"^^xsd:nonNegativeInteger] .`

The Features of OWL

- OWL also allows us to encode the **following property restrictions**
  - `owl:qualifiedcardinality`, `owl:maxQualifiedCardinality`, and `owl:minQualifiedCardinality`
- **Take home message**, we can define all kinds of constraints on how classes and properties should be understood with respect to other classes and properties -- these can be **automatically checked**
- How do we get computers to check these are valid in a particular ontology?

OWL Tools

- There exist numerous **reasoners** (FaCT++, HermiT, Pellet, and Racer) and **ontology editors** including the **NeOn** toolkit, **TopBraid** (a commercial product) and the **Fluent Editor** (which uses Controlled Natural Language in its interface) for OWL
- To date the most popular tool for OWL is Stanford University's **Protégé**, a free open source ontology editor
- The reasoners mentioned above are included with Protégé

Closed vs Open World

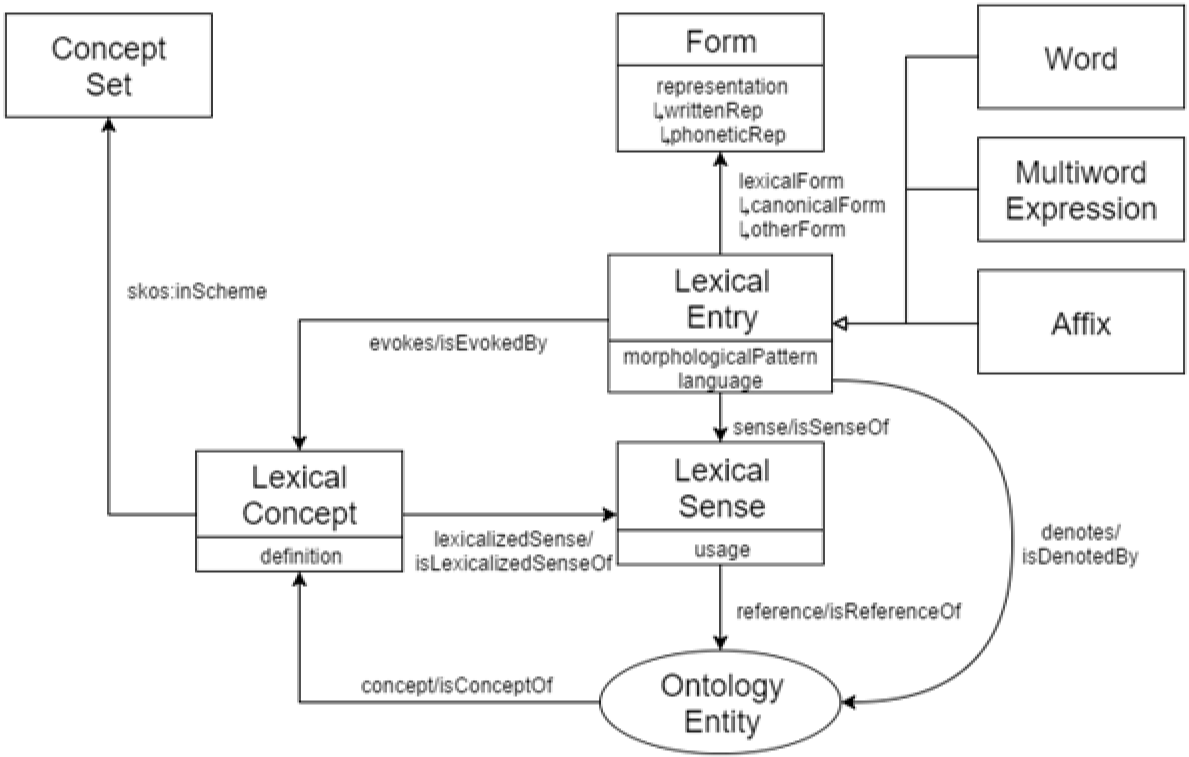
- OWL and (many other knowledge engineering languages) make the **open world assumption**, and it's important to know what this is.
- It basically means that **the failure to derive a statement from our ontology does not allow us to assume its falsity**
    - E.g., if we can only derive one instance of the relation hasChild with Donald as subject, i.e., hasChild(donald, ivanka), this doesn't mean that Donald has only one child...*unless* we have made this explicit elsewhere in the ontology

Closed vs Open World

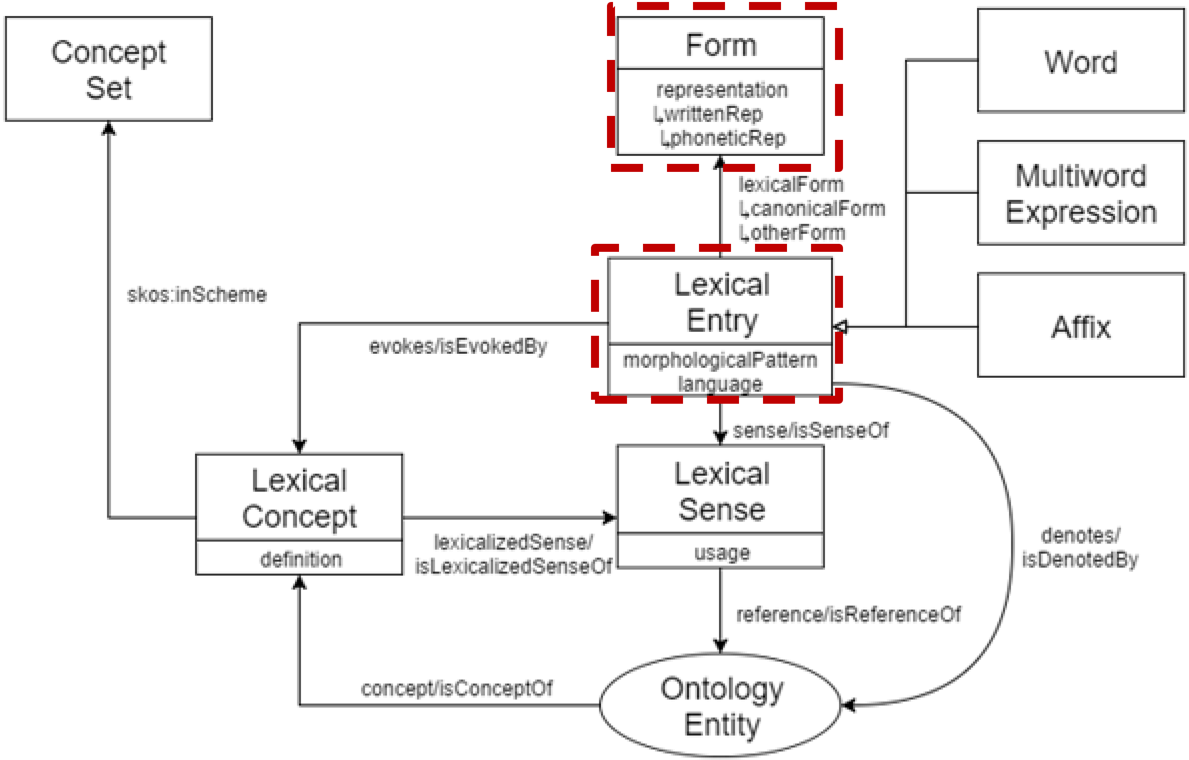
- The **closed world assumption** would allow us to make this inference without further knowledge; databases usually make this assumption for example, neither does the logic programming language PROLOG
- This assumption means that reasoning with OWL **isn’t always as intuitive as we would like** and sometimes its better to find additional ways of adding constraints on RDF datasets that can be automatically checked
- One increasing popular way to do this is to use the **Shapes Query Language (SHACL)**
    - But this would take us too far off topic!

Domain Ontologies: OntoLex-Lemon

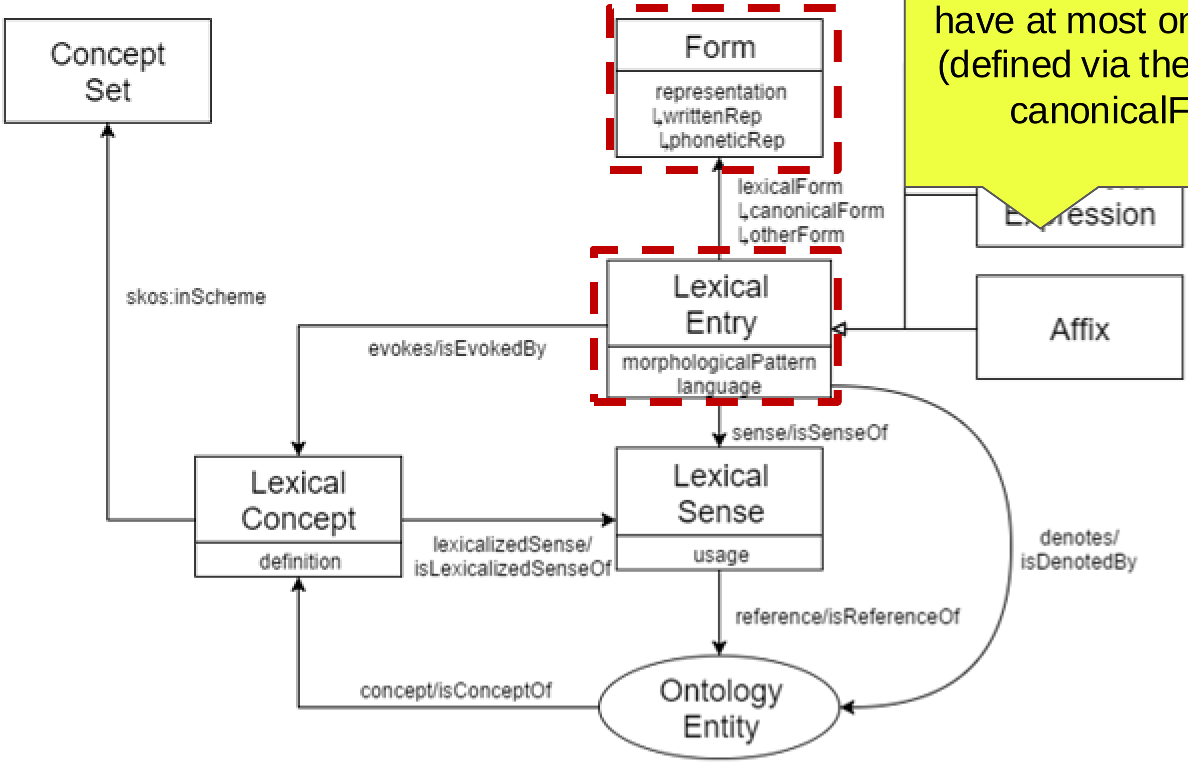
- We have already looked at some top level ontologies, one interesting example of a **domain ontology** is OntoLex-Lemon, used for creating lexical resources.
- OntoLex is a **modular ontology** that includes a **core** module as well as others, which we will look at in detail in the next lesson (where I will show how to use the different classes and properties it contains)
- In this part of the lesson I want to look at OntoLex core module as a **resource**, as an **OWL ontology**
- We will see how the core is defined using the OWL language, using some of the OWL constructs we have just seen
- This is useful in order to know how to re-use the classes and properties in OntoLex

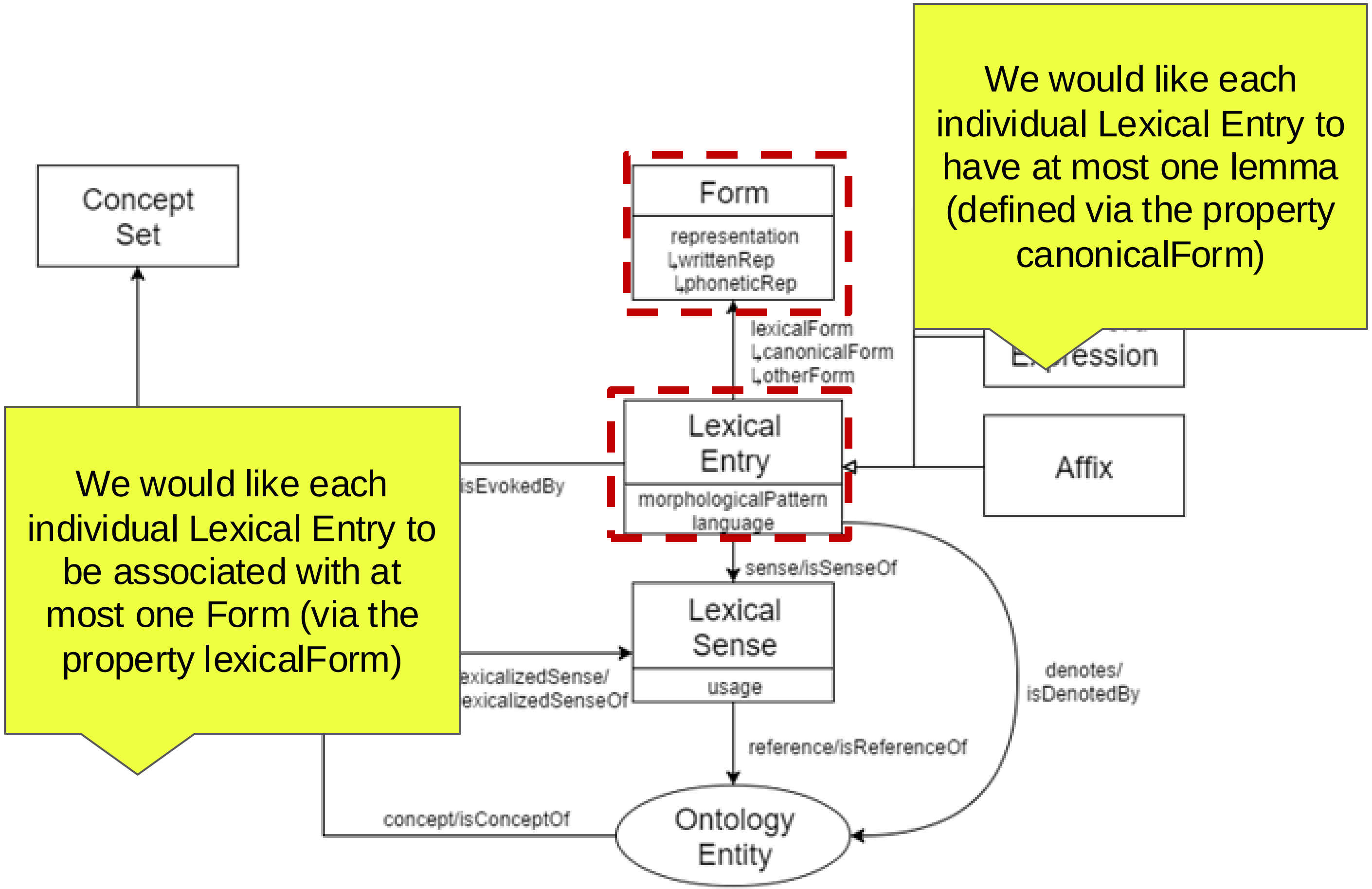


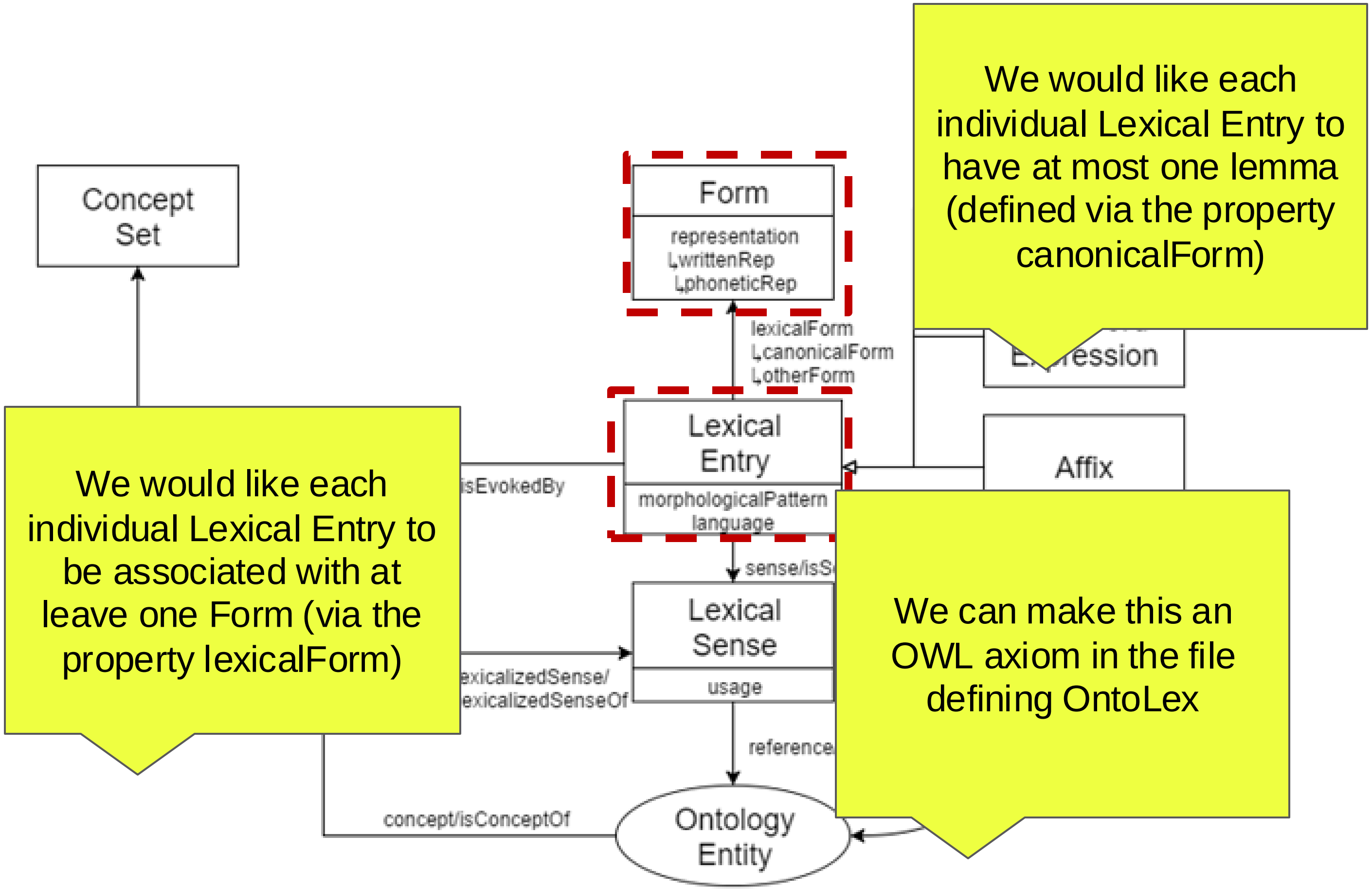
Concept Set
Form
representation
↳writtenRep
↳phoneticRep
Word
Multiword Expression
Affix
lexicalForm
↳canonicalForm
↳otherForm
skos:inScheme
Lexical Entry
morphologicalPattern
language
evokes/isEvokedBy
sense/isSenseOf
Lexical Concept
definition
Lexical Sense
usage
lexicalizedSense/
isLexicalizedSenseOf
denotes/
isDenotedBy
reference/isReferenceOf
concept/isConceptOf
Ontology Entity


Concept Set
skos:inScheme
Form
representation
↳writtenRep
↳phoneticRep
lexicalForm
↳canonicalForm
↳otherForm
Word
Multiword Expression
Affix
Lexical Entry
morphologicalPattern
language
evokes/isEvokedBy
sense/isSenseOf
Lexical Concept
definition
lexicalizedSense/
isLexicalizedSenseOf
Lexical Sense
usage
denotes/
isDenotedBy
reference/isReferenceOf
concept/isConceptOf
Ontology Entity

We would like each individual Lexical Entry to have at most one lemma (defined via the property canonicalForm)



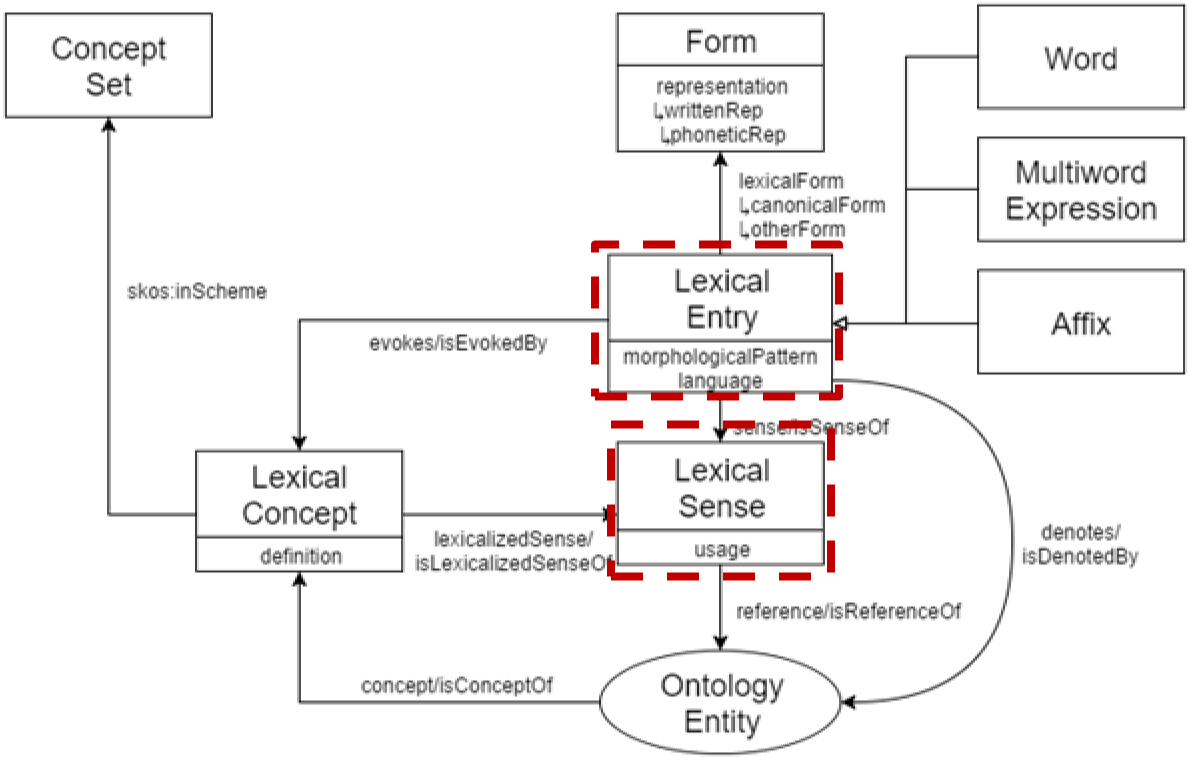

Concept Set
Form
representation
writtenRep
phoneticRep
lexicalForm
canonicalForm
otherForm
We would like each individual Lexical Entry to have at most one lemma (defined via the property canonicalForm)
Lexical Entry
morphologicalPattern
language
Affix
isEvokedBy
We would like each individual Lexical Entry to be associated with at most one Form (via the property lexicalForm)
sense/isSenseOf
Lexical Sense
usage
denotes/
isDenotedBy
reference/isReferenceOf
concept/isConceptOf
Ontology Entity

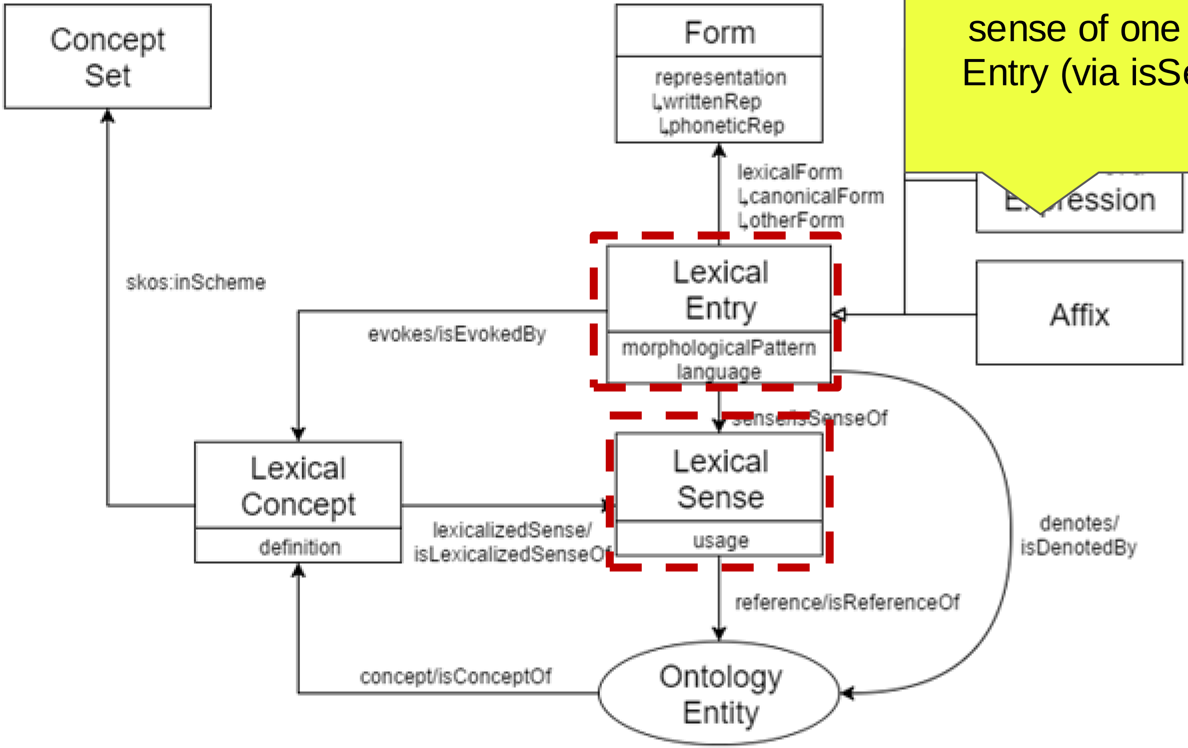


We would like each individual Lexical Entry to have at most one lemma (defined via the property canonicalForm)
Concept Set
Form
representation
writtenRep
phoneticRep
lexicalForm
canonicalForm
otherForm
Lexical Entry
morphologicalPattern
language
isEvokedBy
Affix
We would like each individual Lexical Entry to be associated with at leave one Form (via the property lexicalForm)
Lexical Sense
usage
We can make this an OWL axiom in the file defining OntoLex
concept/isConceptOf
Ontology Entity

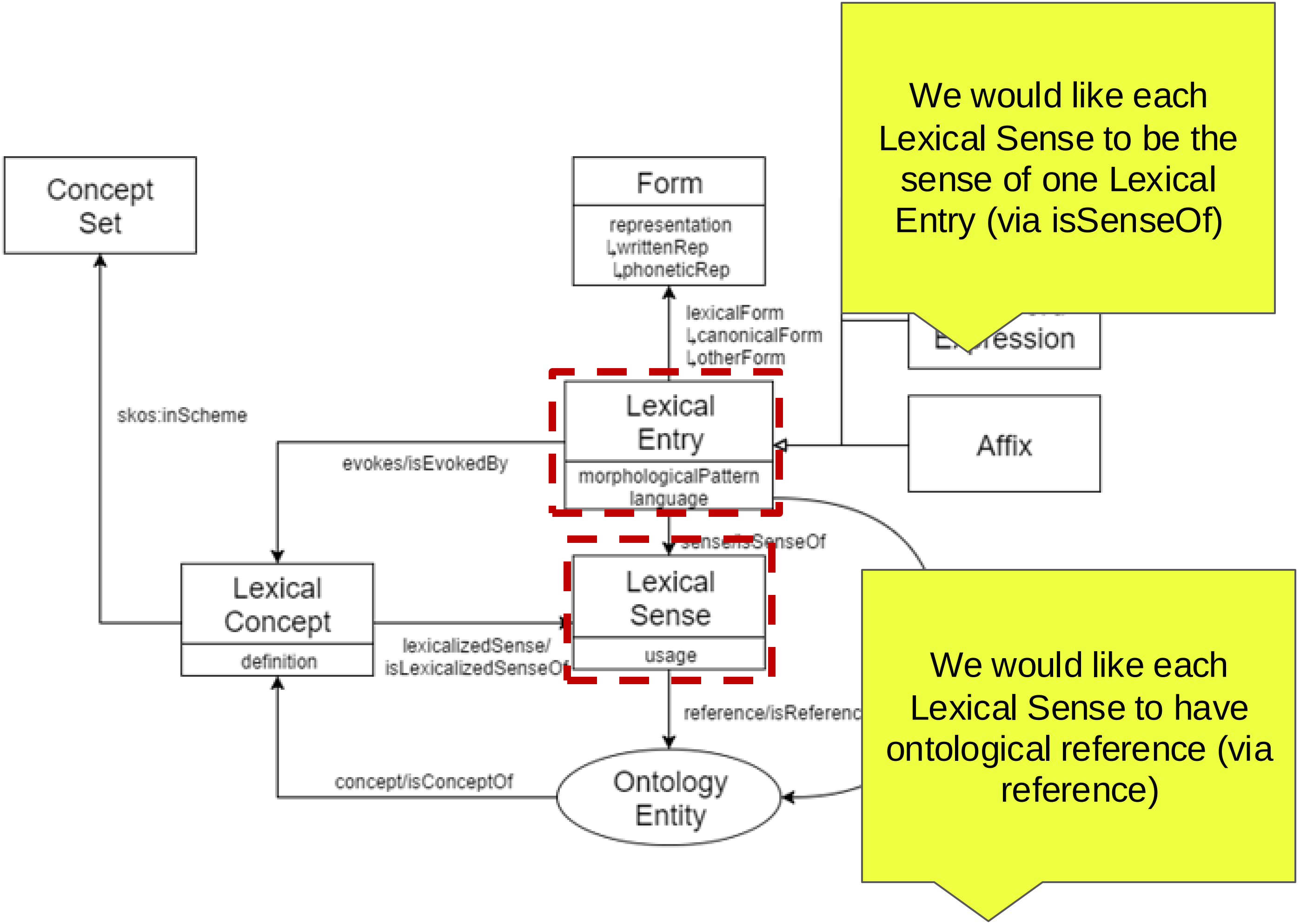
```
<http://www.w3.org/ns/lemon/ontolex#LexicalEntry> a owl:Class ;

  rdfs:label "leksikale inskrywing"@af, "Lexikoneintrag"@de, "lexical entry"@en, "словарная единица"@ru ;

  rdfs:subClassOf owl:Thing, [ a owl:Restriction ;

    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#canonicalForm> ;

    owl:onClass <http://www.w3.org/ns/lemon/ontolex#Form> ;

    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger

  ], [ a owl:Restriction ;

    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#lexicalForm> ;

    owl:onClass <http://www.w3.org/ns/lemon/ontolex#Form> ;

    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger

  ] ;

  rdfs:comment "A lexical entry represents[…]."@en;  rdfs:isDefinedBy <http://www.w3.org/ns/lemon/ontolex> .
```

```
<http://www.w3.org/ns/lemon/ontolex#LexicalEntry> a owl:Class ;

  rdfs:label "leksikale inskrywing"@af, "Lexikoneintrag"@de, "lexical entry"@en, "словарная
единица"@ru ;

  rdfs:subClassOf owl:Thing, [ a owl:Restriction ;

    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#canonicalForm> ;

    owl:onClass <http://www.w3.org/ns/lemon/ontolex#Form> ;

    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger

  ], [ a owl:Restriction ;

    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#lexicalForm> ;

    owl:onClass <http://www.w3.org/ns/lemon/ontolex#Form> ;

    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger

  ] ;

  rdfs:comment "A lexical entry represents[…]."@en;  rdfs:isDefinedBy
<http://www.w3.org/ns/lemon/ontolex> .
```


Concept Set
Form
representation
↳writtenRep
↳phoneticRep
Word
Multiword Expression
Affix
lexicalForm
↳canonicalForm
↳otherForm
skos:inScheme
Lexical Entry
morphologicalPattern
language
evokes/isEvokedBy
sense/isSenseOf
Lexical Concept
definition
Lexical Sense
usage
lexicalizedSense/
isLexicalizedSenseOf
denotes/
isDenotedBy
reference/isReferenceOf
concept/isConceptOf
Ontology Entity



We would like each Lexical Sense to be the sense of one Lexical Entry (via isSenseOf)

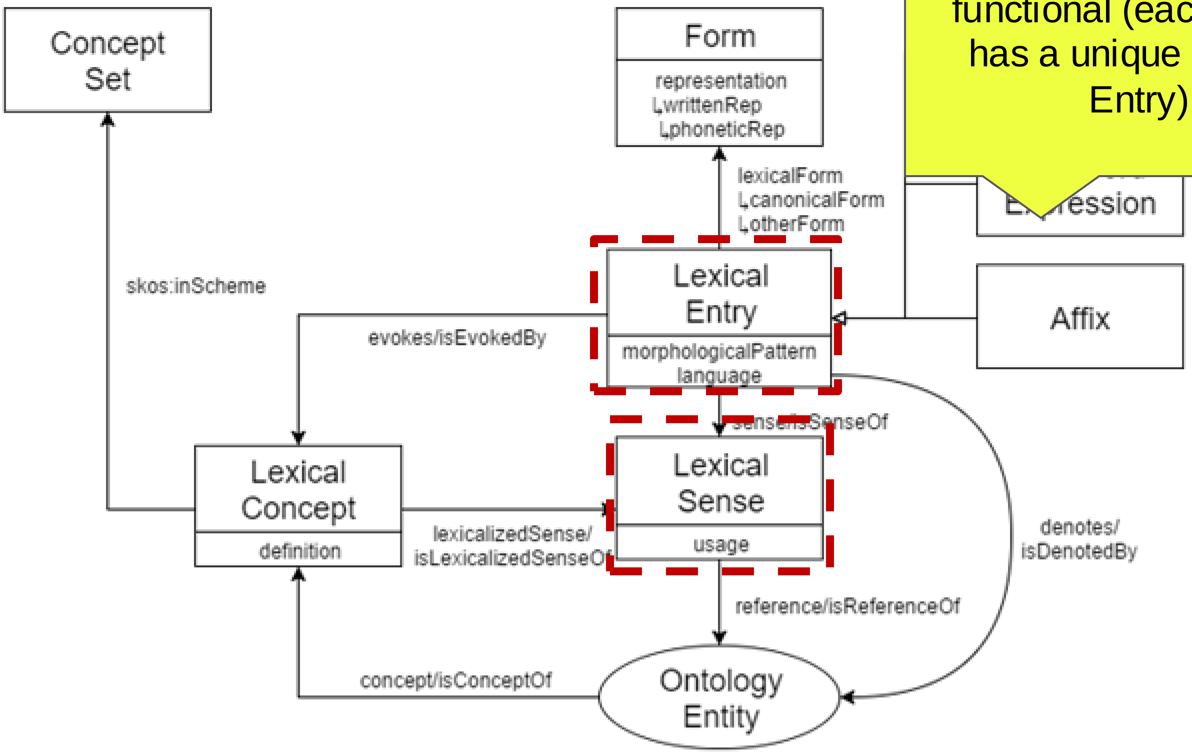


We would like each Lexical Sense to be the sense of one Lexical Entry (via isSenseOf)
Concept Set
Form
representation
writtenRep
phoneticRep
lexicalForm
canonicalForm
otherForm
skos:inScheme
Lexical Entry
morphologicalPattern
language
Affix
evokes/isEvokedBy
Lexical Concept
definition
lexicalizedSense/
isLexicalizedSenseOf
Lexical Sense
usage
concept/isConceptOf
Ontology Entity
We would like each Lexical Sense to have ontological reference (via reference)

```
<http://www.w3.org/ns/lemon/ontolex#LexicalSense>

  a owl:Class ;

  rdfs:label "leksikale sin"@af, "lexikalischer Sinn"@de, "lexical sense"@en, "лексический
смысл"@ru ;

 rdfs:subClassOf owl:Thing, [ a owl:Restriction ;

    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#isSenseOf> ;

    owl:onClass <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> ;

    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger

  ], [

    a owl:Restriction ;

    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#reference> ;

    owl:cardinality "1"^^xsd:nonNegativeInteger

  ] ;

  rdfs:comment "A lexical sense represents […]"@en ;

  rdfs:isDefinedBy <http://www.w3.org/ns/lemon/ontolex> .
```

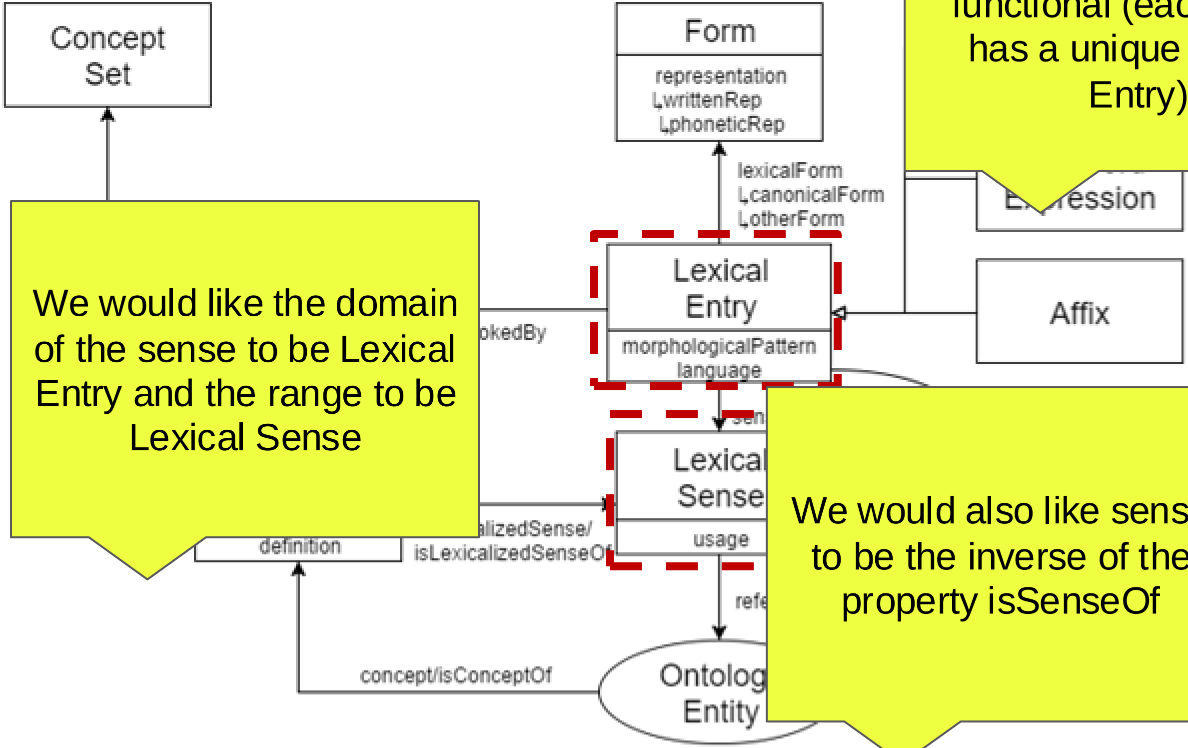
```
<http://www.w3.org/ns/lemon/ontolex#LexicalSense>
  a owl:Class ;
  rdfs:label "leksikale sin"@af, "lexikalischer Sinn"@de, "lexical sense"@en, "лексический смысл"@ru ;
 rdfs:subClassOf owl:Thing, [ a owl:Restriction ;
    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#isSenseOf> ;
    owl:onClass <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
  ], [
    a owl:Restriction ;
    owl:onProperty <http://www.w3.org/ns/lemon/ontolex#reference> ;
    owl:cardinality "1"^^xsd:nonNegativeInteger
  ] ;
  rdfs:comment "A lexical sense represents […]"@en ;
  rdfs:isDefinedBy <http://www.w3.org/ns/lemon/ontolex> .
```

We would like the sense property to be inverse functional (each sense has a unique Lexical Entry)



We would like the sense property to be inverse functional (each sense has a unique Lexical Entry)

We would like the domain of the sense to be Lexical Entry and the range to be Lexical Sense



We would also like sense to be the inverse of the property isSenseOf

```
<http://www.w3.org/ns/lemon/ontolex#sense>

  a owl:ObjectProperty, owl:InverseFunctionalProperty ;

  rdfs:label "sinne"@af, "Sinn"@de, "sense"@en, "acepción"@es, "signification"@fr, "senso"@it, 
"zin"@nl, "sentido"@pt, "sens"@ro, "betydelse"@sv, "смысл"@ru ;

  rdfs:comment "The 'sense' property relates a lexical entry to one of its lexical senses.
"@en, "Свойство 'смысл' связывает словарную единицу с одним из ее лексических смыслов."@ru ;

  rdfs:isDefinedBy <http://www.w3.org/ns/lemon/ontolex> ;

  rdfs:domain <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> ;

  rdfs:range <http://www.w3.org/ns/lemon/ontolex#LexicalSense> ;

  owl:inverseOf <http://www.w3.org/ns/lemon/ontolex#isSenseOf> .
```

```
<http://www.w3.org/ns/lemon/ontolex#sense>

  a owl:ObjectProperty, owl:InverseFunctionalProperty ;

  rdfs:label "sinne"@af, "Sinn"@de, "sense"@en, "acepción"@es, "signification"@fr, "senso"@it,
"zin"@nl, "sentido"@pt, "sens"@ro, "betydelse"@sv, "смысл"@ru ;

  rdfs:comment "The 'sense' property relates a lexical entry to one of its lexical senses.
"@en, "Свойство 'смысл' связывает словарную единицу с одним из ее лексических смыслов."@ru ;

  rdfs:isDefinedBy <http://www.w3.org/ns/lemon/ontolex> ;

  rdfs:domain <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> ;

  rdfs:range <http://www.w3.org/ns/lemon/ontolex#LexicalSense> ;

  owl:inverseOf <http://www.w3.org/ns/lemon/ontolex#isSenseOf> .
```

```
<http://www.w3.org/ns/lemon/ontolex#sense>

  a owl:ObjectProperty, owl:InverseFunctionalProperty ;

  rdfs:label "sinne"@af, "Sinn"@de, "sense"@en, "acepción"@es, "signification"@fr, "senso"@it,
"zin"@nl, "sentido"@pt, "sens"@ro, "betydelse"@sv, "смысл"@ru ;

  rdfs:comment "The 'sense' property relates a lexical entry to one of its lexical senses.
"@en, "Свойство 'смысл' связывает словарную единицу с одним из ее лексических смыслов."@ru ;

  rdfs:isDefinedBy <http://www.w3.org/ns/lemon/ontolex> ;

  rdfs:domain <http://www.w3.org/ns/lemon/ontolex#LexicalEntry> ;

  rdfs:range <http://www.w3.org/ns/lemon/ontolex#LexicalSense> ;

  owl:inverseOf <http://www.w3.org/ns/lemon/ontolex#isSenseOf> .
```

Summary and Take Away Message

- **Take away message** (even if you don't quite follow the technical details): OWL can be used to describe the meaning of different classes and properties in an ontology in a machine actionable way.
- Further details on OWL and how to use it can be found in the Protégé pizza tutorial, which exists in several versions, I like this updated one:
    - https://www.michaeldebellis.com/post/new-protege-pizza-tutorial