UNIVERSITA' DEGLI STUDI DI ROMA

TorVergata Università

Thesauri building with SKOS

Armando Stellato,
University of Rome, Tor Vergata

2010 International Symposium on
Agricultural Ontology Services
Beijing – 30-31 October 2010

Outline

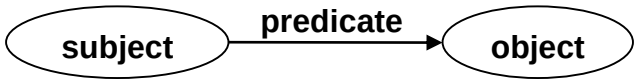
- A Web of Data…: a brief historical introduction.
- …data…and Concepts?
- From data modeling to concepts modeling: SKOS
- Resources for SKOS manipulation
  - Tools
  - Software Libraries
  - Services
- A Demo of a SKOS/OWL Development Environment: Semantic Turkey

A Web of Data

Ontology Languages: a "warp speed" resume (1)

RDF Data Model:

- Deals with representation of resources on the web:
  - "Everything is a resource"
  - An RDF model is a set of statement of the type:
    - Subject – predicate – Object
    - Subject is always a resource, Object can be a value (a simple datatype) or a resource too
    - Predicate is an attributive (for datatypes) / relational (when pointing to resources) property of the subject
    - Even statements can be treated as resources
  - An RDF model can be seen as a labeled directed graph, with each triple:



  - Meaning of a RDF graph: it is the conjunction of all its statements

Armando Stellato
stellato@info.uniroma2.it ai-nlp.info.uniroma2.it/stellato

Ontology Languages: a “warp speed” resume (2)

RDFS extends RDF with a vocabulary for defining knowledge schemas:

- Class, Property
- type, subClassOf, subPropertyOf
- range & domain constraints

OWL (Web Ontology Language), extends RDFS with:

- Contextualized contraints (Person: ∀ has_child.Person
  Elephant: ∀ has_child.Elephant)
- Existential/Cardinality contraints (Parent ⊑ has_child ≥1)
- Property facets (*transitive*, *symmetric*, *inverse* properties…)
- OWL Semantics are based on Description Logics { $SHOIN(D_n)$ }
- OWL 2… {$SROIQ(D_n)$}

Armando Stellato
stellato@info.uniroma2.it ai-nlp.info.uniroma2.it/stellato

Summing up…

- *RDF* provides a modeling infrastructure for representing linked resources
  - Actually, it recalls '60's Semantic Networks…with no Semantics ☺
- *RDF(S)* and *OWL*, provide semantics for RDF
- *They provide schema for organizing data*
  - (Classes are collections of objects, properties characterize data)
- *Support for Inference*
  - trade-off: expressive power vs computational requirements (completeness and decidability)

Armando Stellato
stellato@info.uniroma2.it ai-nlp.info.uniroma2.it/stellato

Accomplished objectives

UNIVERSITA' DEGLI STUDI DI ROMA TorVergata Università

Two birds with one stone!

Replacing 80s relation model (DBs)

- Closer to human understandability (reminds of ER diagrams!)
- With well-founded logical ground

Putting data on the Web!

Armando Stellato
stellato@info.uniroma2.it art.uniroma2.it/stellato

UNIVERSITA' DEGLI STUDI DI ROMA

TorVergata Università

A Web of Data

…and what about Concepts?

Do we need anything else?

So, ontologies, in a certain sense, replace those old fashioned DB tables and constraints

Though, these data schemata:

- scale better!
  - try to manage hundreds of interconnected tables…
  - have your domain expert add a new entity in the middle of an entity tree in the ER, and then try to reengineer the DB schema
- are better understandable
- are better shareable
  - Try to merge two DB schema…

[1] "a la" Guarino, that is, separated from instance data, or: Terminology Boxes in Description Logics dialect

Do we need anything else?

With such a rich set of KR languages…wouldn't be that easy to develop dictionaries/thesauri?

- Thesauri are simpler than ontologies!

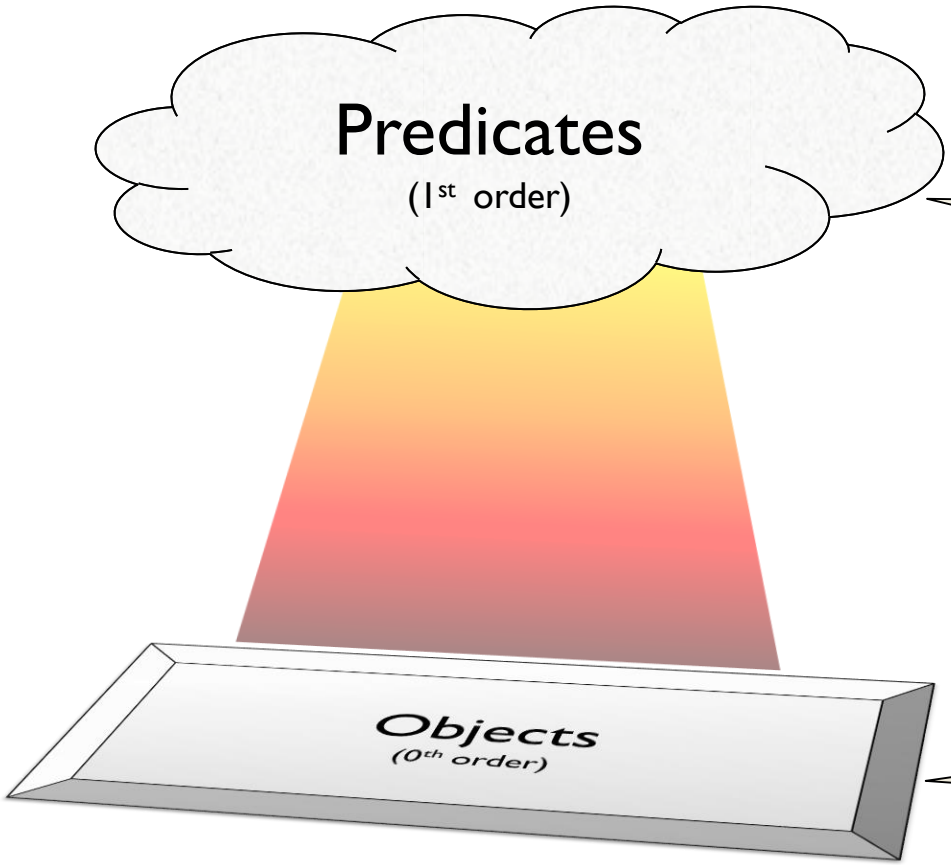
- RDF/RDFS/OWL allow for:
  - Concept Hierarchies
  - Description of concepts through properties
  - That's all we need!

Maybe yes…

With such a rich set of KR languages…wouldn't be that easy to develop dictionaries/thesauri?

- With DL semantics applied to data schema…you bought:
  - heavy restrictions
  - commitment
- Description logics are restrictions of 1° order logic
  - Not able to predicate over predicates…
- Classification Issues:
  - What happens when concept = class?

First order logics



logically "describe" objects of the domain

Cannot be described themselves (unless through 2nd order predicates)

logically "described" by predicates

Is an Ontology Language good for Thesauri?

concept = owl:Class?

rdfs:subClassOf used for the hierarchy?

then…

- Not able to characterize concepts (need 2nd order, remember?)
- Do we need instances? (0th order, if not, we just need to go down one level ☺)

So…probably not if used as a "first-glance" would suggest…we need something else…

Are Thesauri good for Ontologies?

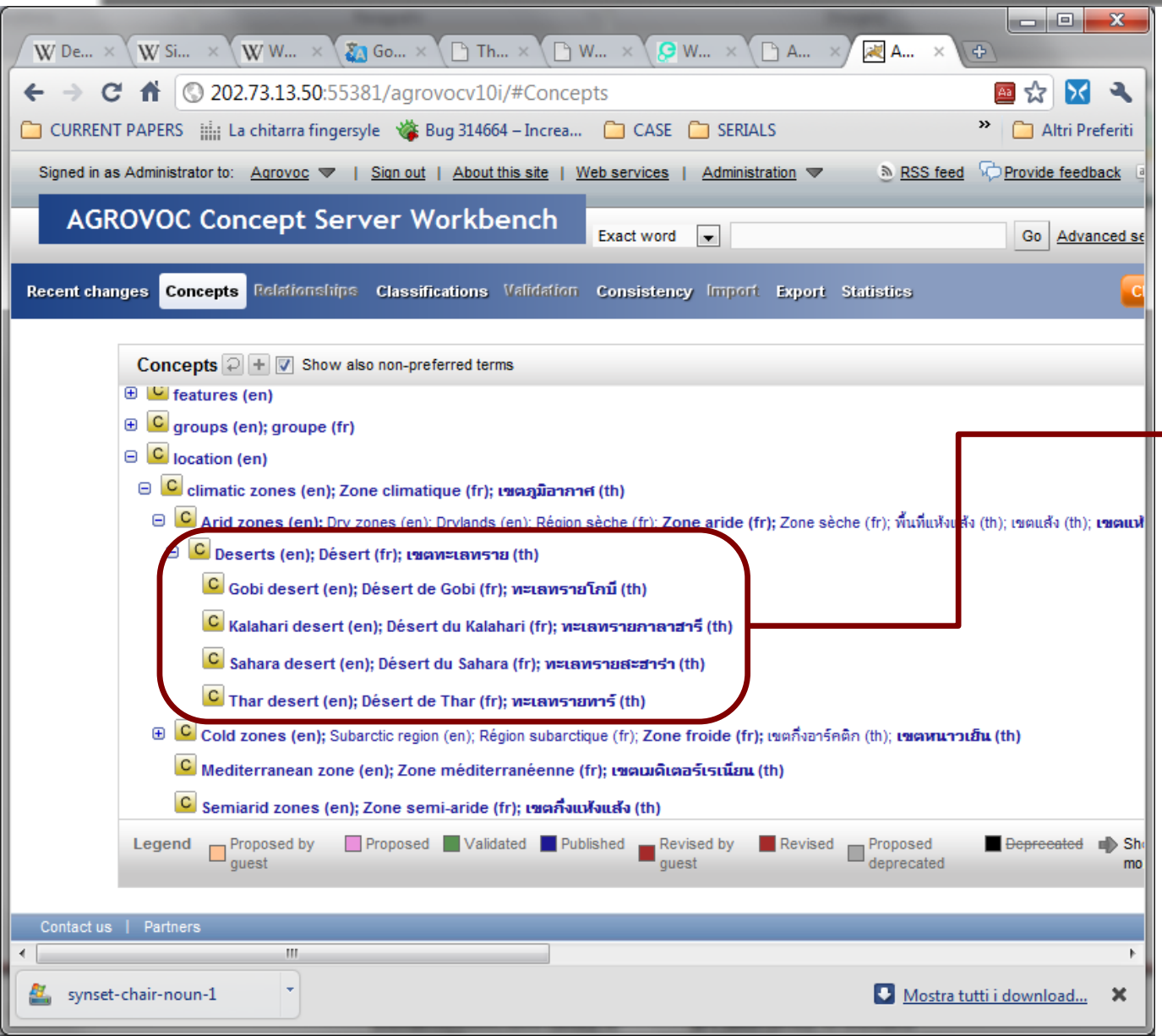
Tempting to reuse all the information from available knoweldge resources

But misuse is round the corner!

- Formal semantic consistency of reused concepts difficult to assure for very large thesauri
- Concept/instance separation? At least some clean up is necessary…

Are Thesauri good for Ontologies?

UNIVERSITA' DEGLI STUDI DI ROMA
Tor Vergata
Università



The generic broader/narrower relationship may hold between Arid Zones and Deserts, and between Deserts and:

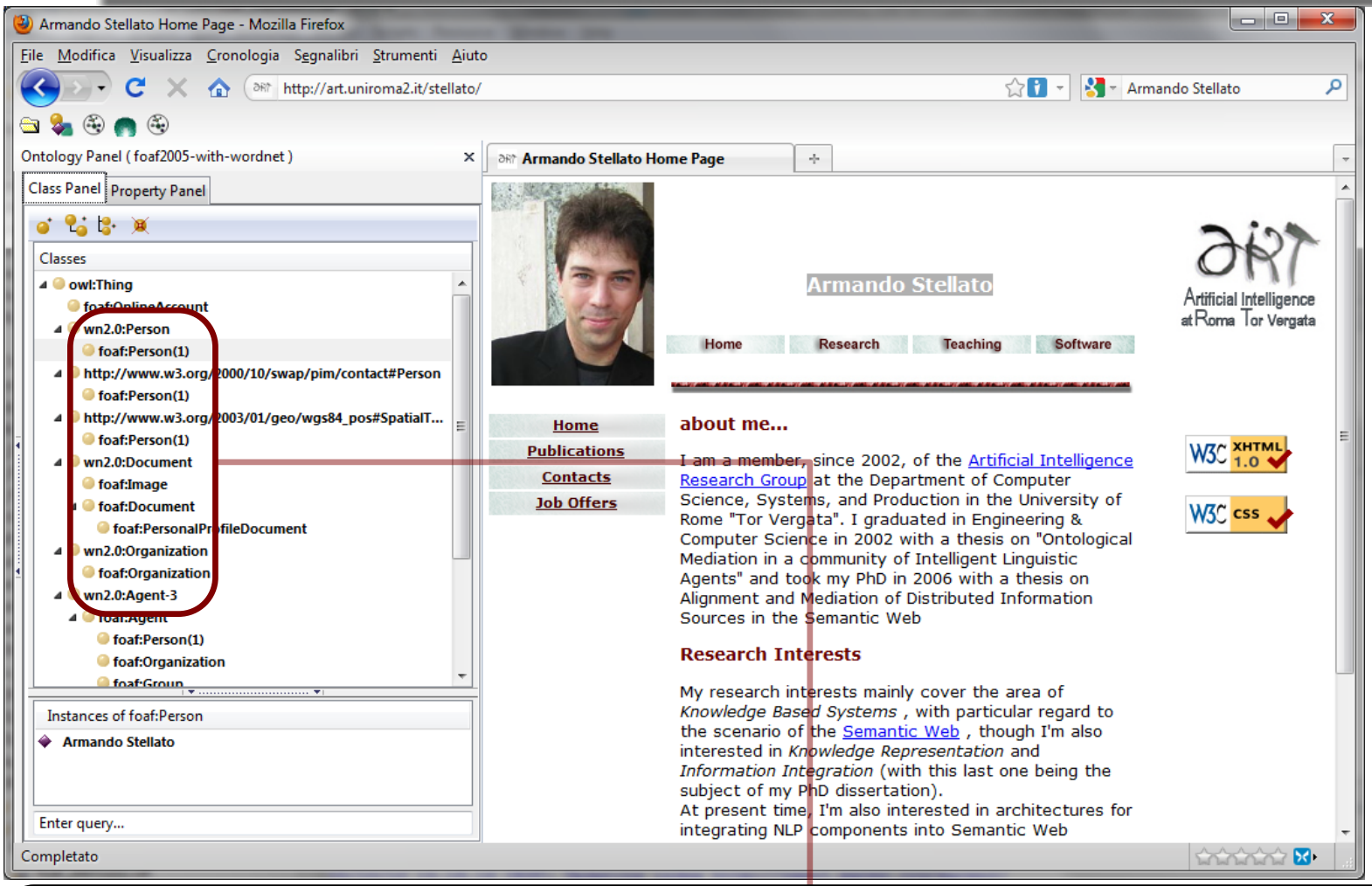
- Gobi Desert
- Kalahari Desert
- Sahara Desert
- Thar Desert

But, ontologically, here we have one (or even two) jumps of logical order!

Ex: Reuse of thesauri as ontologies

first W3C WordNet RDF, used in FOAF

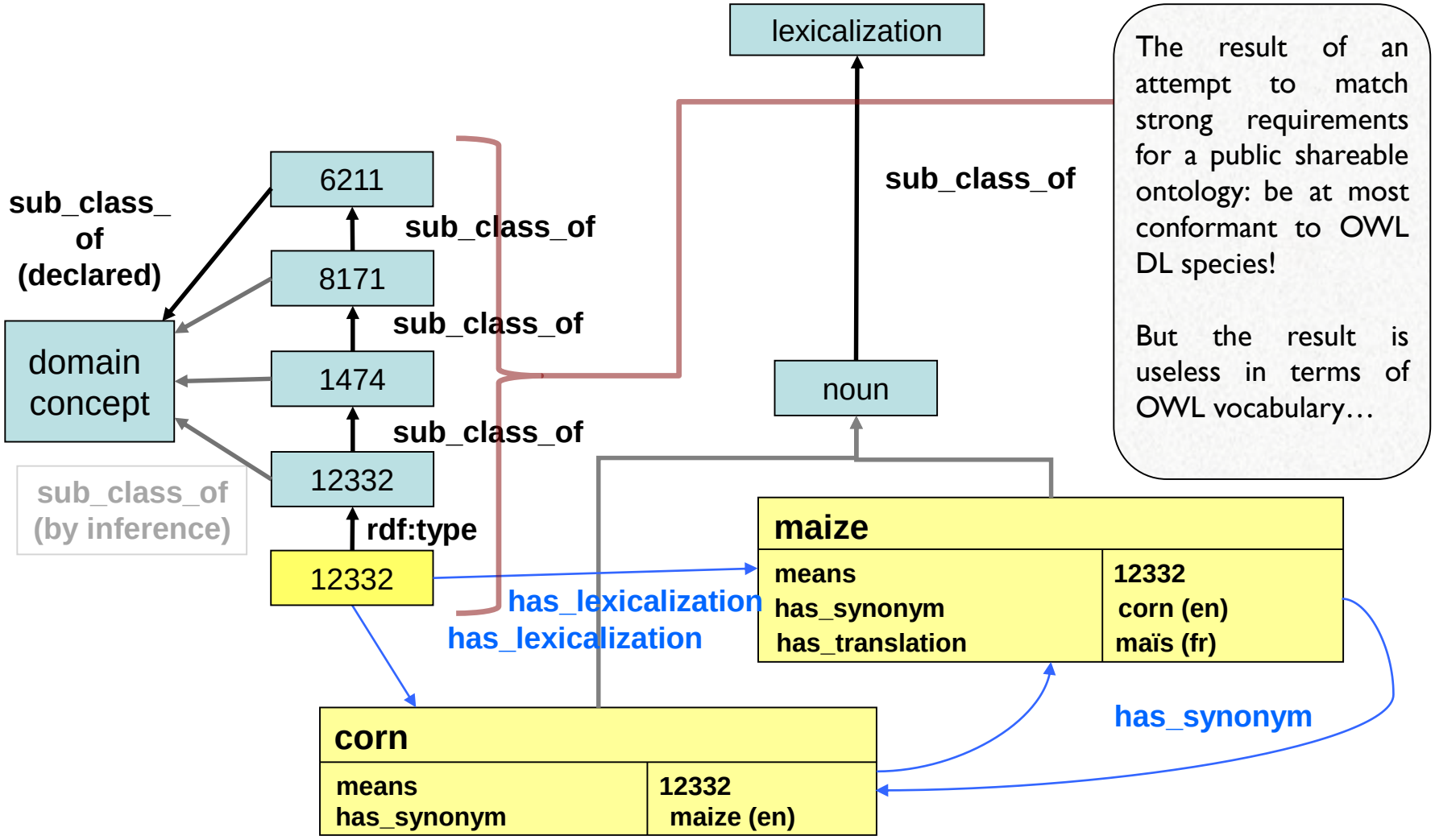
UNIVERSITA' DEGLI STUDI DI ROMA Tor Vergata Università



Still a dedicated formalization has been made necessary!

WordNet has been first ported to RDF in 2005 as an OWL ontology, with synset mapped as classes. It has also being linked by the 2005 version of the FOAF ontology.
Then in 2006 (Van Assem, Gangemi, Schreiber) a dedicated WordNet task-force re-interpreted it still as an OWL ontology, but as an ontology of **language** rather than **domain**. Today there's a mapping of WordNet under the umbrella of the Ontolex/Lemon lexicon model

Another Example
Agrovoc as it was modeled in OWL



UNIVERSITA' DEGLI STUDI DI ROMA

Tor Vergata Università

From data modeling to concepts modeling:

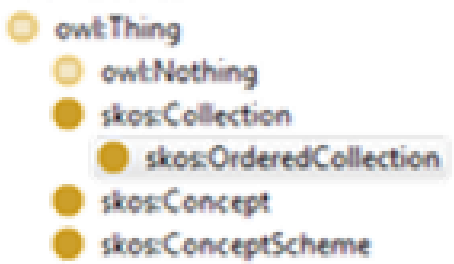
Simple Knowledge Organization Systems

SKOS

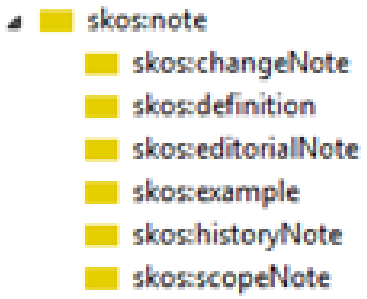
- Move everything one down logical layer!
  - speak *about* concepts, not *using* them to speak about objects
- Lose strong semantic assumptions
  - Loose semantic relations
    - Intra-scheme (narrower/broader)
    - Extra scheme (matching properties vs owl:sameAs/equivalentClass/Property)
- Improved vocabulary for:
  - Codification
  - Language: better descriptions, Internazionalization etc..

SKOS Features for Thesauri

- Short OWL vocabulary, describing SKOS resources



- Support for different Views, through skos:ConceptSchemes
- Support for key identifiers (skos:notations)
- Better characterization of labels:



- Dedicated vocabulary for concept documentatic

SKOS Integrity Conditions

SKOS has several integrity conditions, though they cannot be specified as OWL contraints (mostly property disjointness[1])

- skos:prefLabel, skos:altLabel and skos:hiddenLabel are pairwise disjoint properties.
- A resource has no more than one value of skos:prefLabel per language tag.
- skos:related is disjoint with the property skos:broaderTransitive.
- skos:exactMatch is disjoint with each of the properties skos:broadMatch and skos:relatedMatch.
- There *should* not be (suggested to avoid as a best practice) two different values x and y of skos:notation so that:
  - $\exists$ s s.t. { s skos:notation x .
    s skos:notation y}
  - datatype(x) == datatype (y)

[1] though in OWL2 it is possible to state disjoint properties

SKOS is not OWL-free!!!

SKOS is not an alternative language disjoint from OWL

- It is an OWL vocabulary!
- Exploits much of OWL reasoning
- Its elements are defined basing on OWL
- Wide use of datatype, object, annotation properties as defined in OWL

SKOS Idiosyncrasies

UNIVERSITA' DEGLI STUDI DI ROMA TorVergata Università

Requires Reasoning!!!

- Narrower/broader
  - At least best practices should advice to use just one (narrower, such as for rdfs:subClassOf)
  - Unless, reasoning is *necessary*, which should not be the case

Seem to be done to avoid large computation, but requires more write-time data management

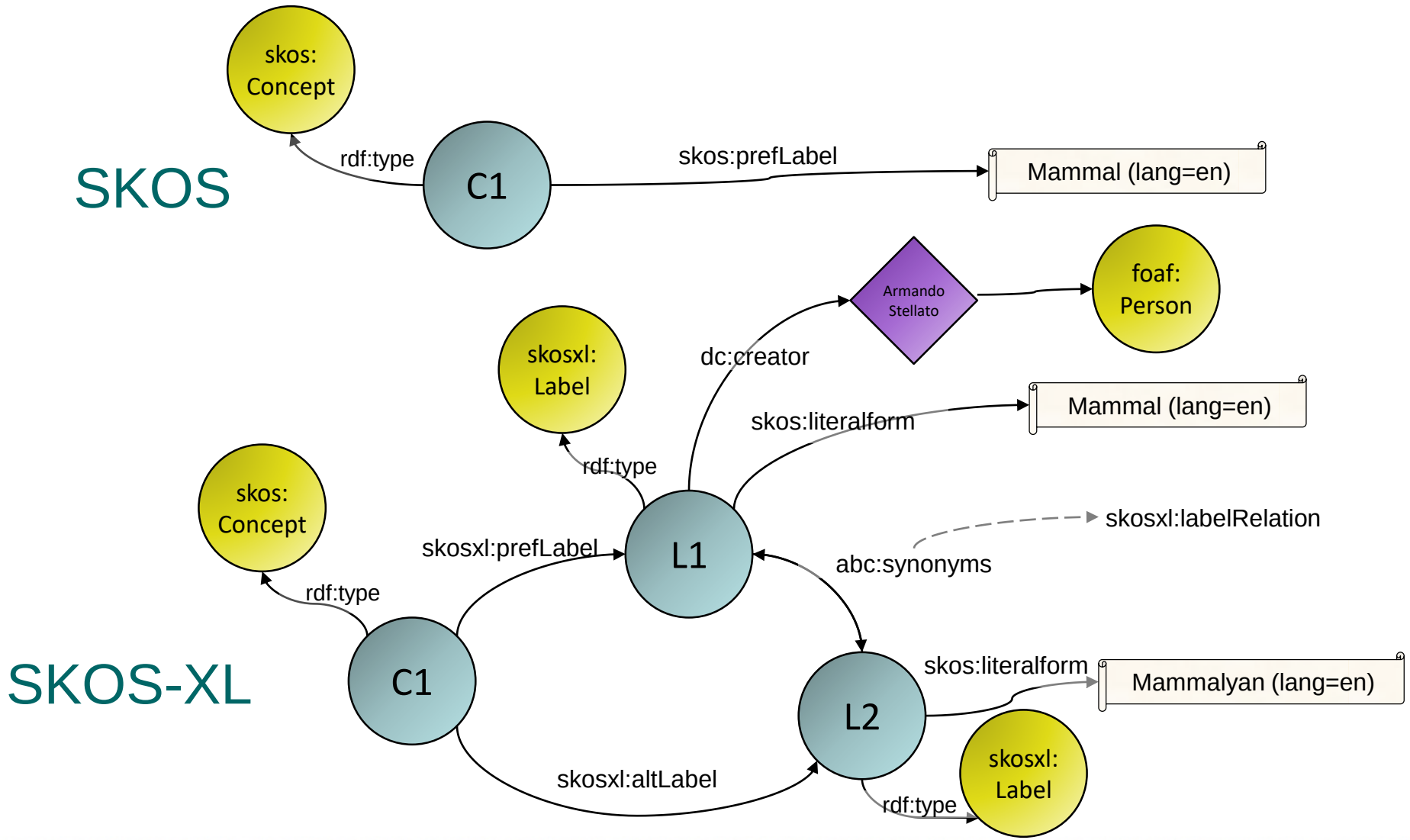
- topConceptOf/hasTopConcept

SKOS-XL

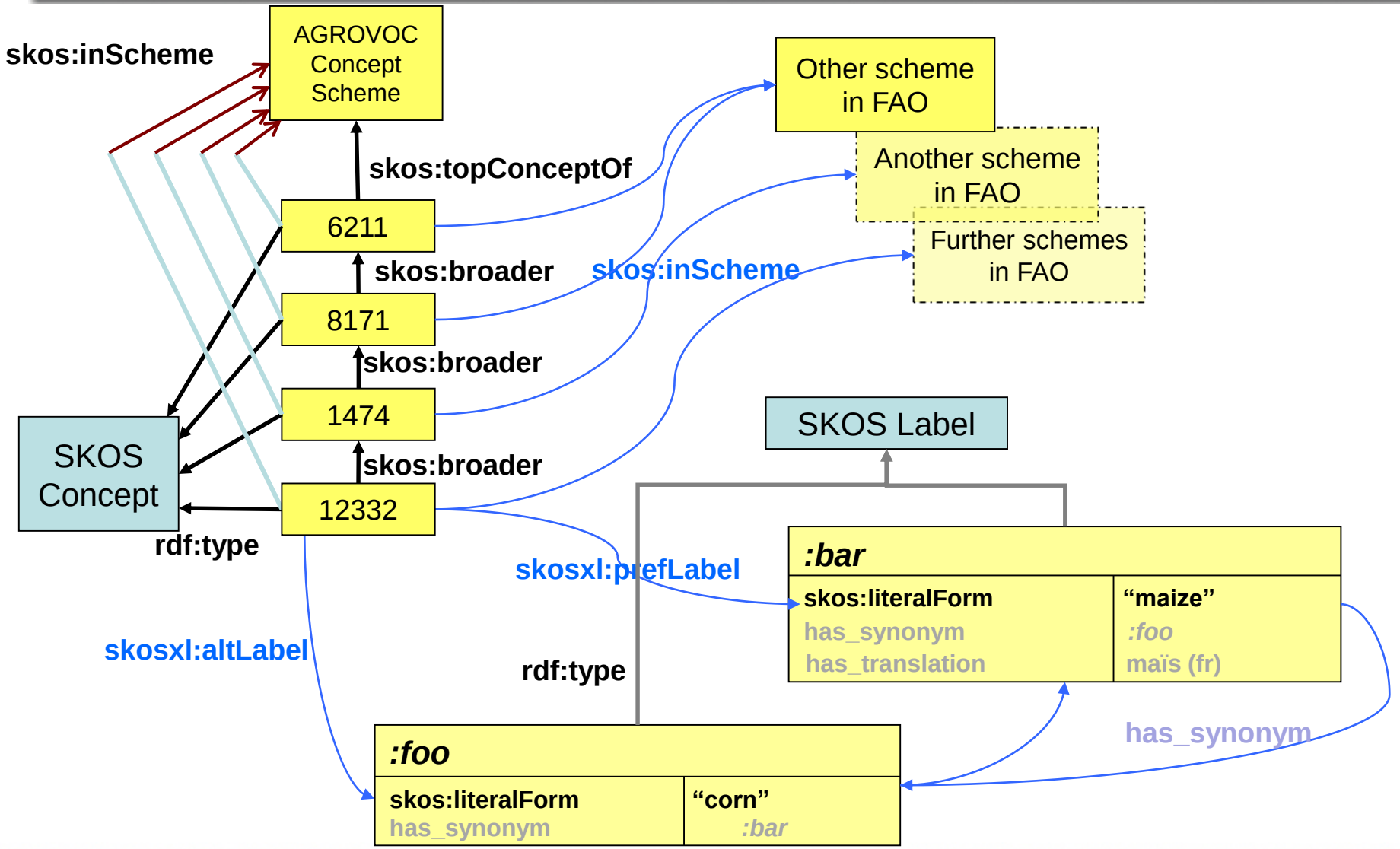
- Thesauri, Dictionaries, Terminologies
  - Often have softer semantics
  - But require richer linguistic characterization!

- Terms/Labels/Synonyms/Translations etc..
  - Need to be reified!
  - I.E. become first class citizens! 0^{th} order objects (much as concepts) which can be described in turn

SKOS-XL


SKOS
skos:
Concept
rdf:type
C1
skos:prefLabel
Mammal (lang=en)
Armando
Stellato
foaf:
Person
dc:creator
skosxl:
Label
skos:literalform
Mammal (lang=en)
rdf:type
skos:
Concept
skosxl:prefLabel
L1
skosxl:labelRelation
abc:synonyms
rdf:type
SKOS-XL
C1
L2
skos:literalform
Mammalyan (lang=en)
skosxl:
Label
skosxl:altLabel
rdf:type

AGROVOC conceptual model, in SKOS-XL



SKOS/SKOS-XL: dangling pointers…

- No relationship between Named Graphs and Schemes…any best practices?
- Which is the intended use for skosxl:Labels?
  - E.g. Should two concepts sharing a lexicalization point to the same skosxl:Label? Shouldn't they?
- Shouldn't SKOS provide default extensions for reifying documentation props too?
- Language aspects: why not providing the definitive vocabulary for this? (linguistic/semantic relationships between terms etc…)

So…

- OWL and SKOS are not enemies!
  - More like father and son ☺
- Mix them up according to what you need, providing that:
  - OWL property axioms may be used freely in any SKOS thesaurus
  - Same concept may be handled as an OWL class and a SKOS concept, but in two different sets of data (linked data) [do not use owl:import!]