Knowledge Representation seminar Meeting #1

Michele Pasin
Kings College, London
June 2010

Outline

- what are ontologies?
  - *[theoretical perspective]*

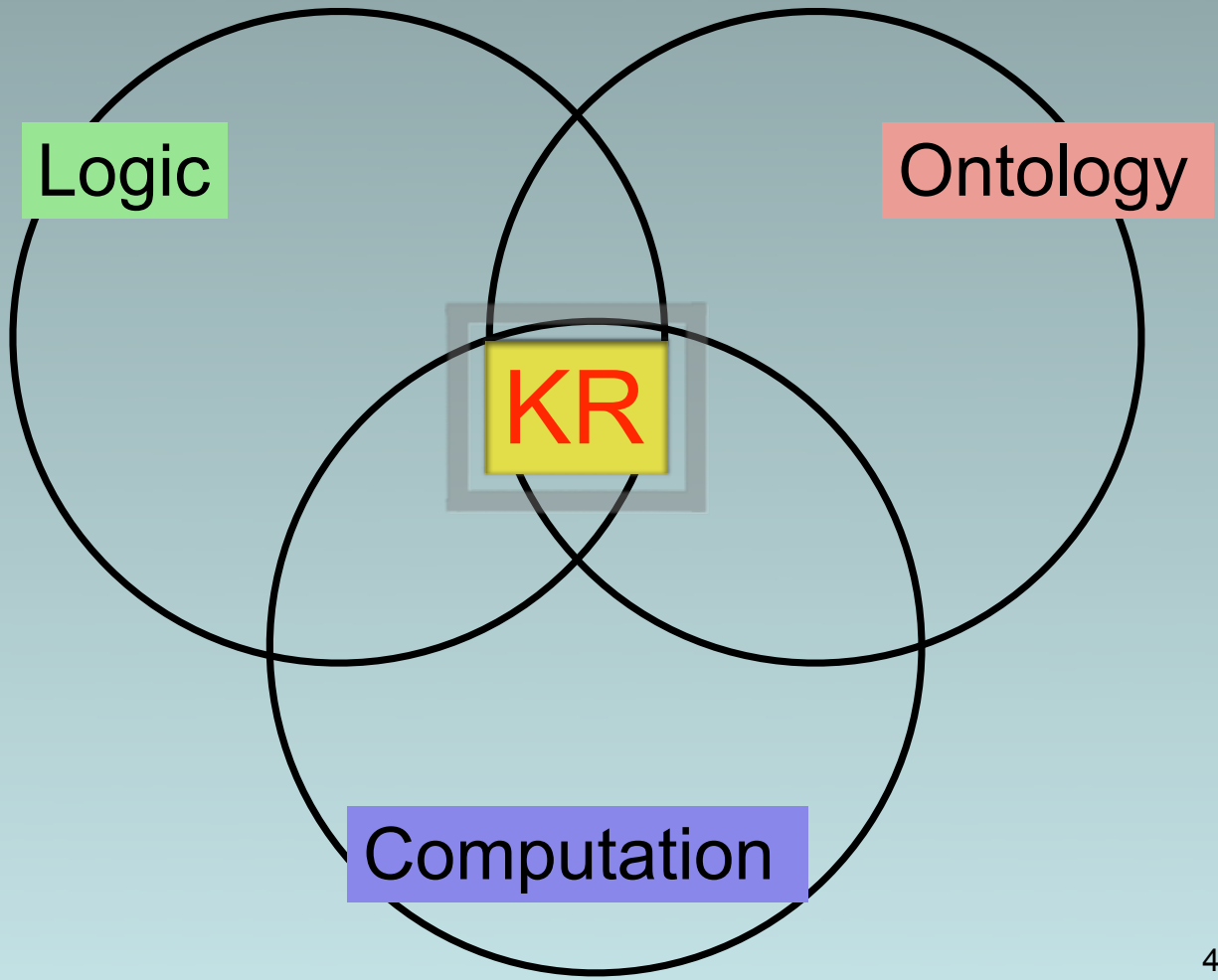
- what are they for?
  - *[pragmatic perspective]*

- how do we build them?
  - *[design perspective]*

What is an ontology? A plethora of definitions..

1. *Ontology as a philosophical discipline*
2. *Ontology as a an informal conceptual system*
3. *Ontology as a formal semantic account*
4. *Ontology as a specification of a conceptualization*
5. *Ontology as a representation of a conceptual system via a logical theory*
   5.1 characterized by specific formal properties
   5.2 characterized only by its specific purposes
6. *Ontology as the vocabulary used by a logical theory*
7. *Ontology as a (meta-level) specification of a logical theory"* (from Guarino & Giaretta, 1995).

Sowa: 3 components to a knowledge representation



Sowa. Knowledge Representation: Logical, Philosophical and Computational Foundations. Course Technology (1999)

(I) Logic

- formal language for expressing the structures used in our inference processes

All x is b.	(Universal Affirmative)
There is a Y that is x.	(Particular Affirmative)
Therefore, y is b.	(Particular Affirmative)



All Roman tribunes have immunity	(Universal Affirmative)
Valerianus is a tribune.	(Particular Affirmative)
Therefore, Valerianus has immunity.	(Particular Affirmative)

(II) Ontology

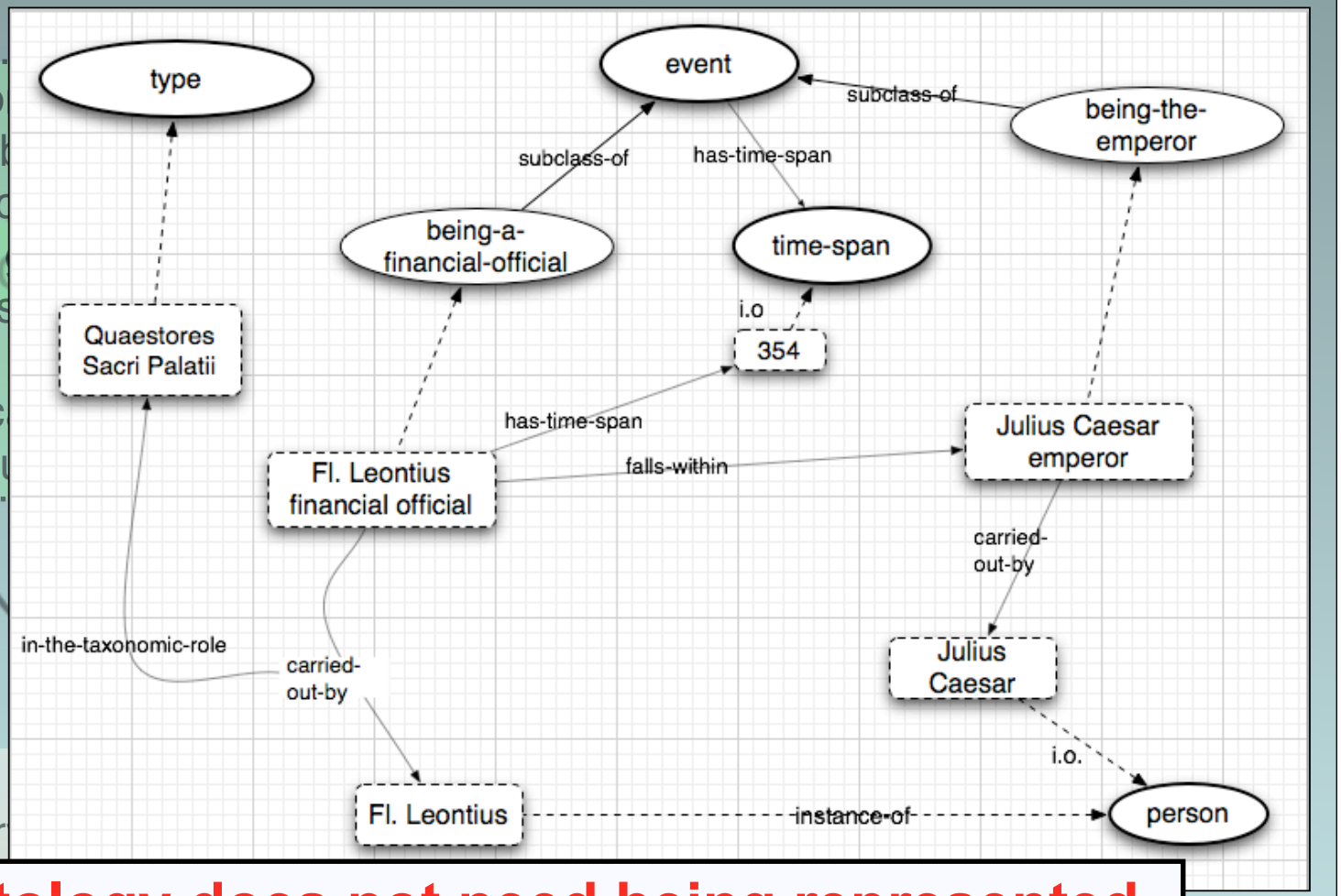
Tribune (from the Latin: tribunus; Byzantine Greek form τριβούνος) was a title shared by **10 elected officials** in the **Roman Republic**. Tribunes had the power to convene the **Plebeian Council** and to act as its **president**, which also gave them the **right to** propose **legislation** before it. They were sacrosanct, in the sense that any **assault** on their person **was prohibited**. They had the power to **veto** actions taken by **magistrates**, and specifically to **intervene legally** on behalf of **plebeians**. The tribune could also summon the **Senate** and **lay proposals** before it. [....]



For every x, if (x **isTribune**) ==> exists y such that (y **isCity**) and (y **hasName** Rome) and (**lives_in** x, y)

(II) Ontology

Tribune (fro
title shared b
the power to
which also
were sacros
prohibited.
and specific
could also su



For ever

- an ontology does not need being represented through the formal language of logic!

(III) Computation

- execution time of a program

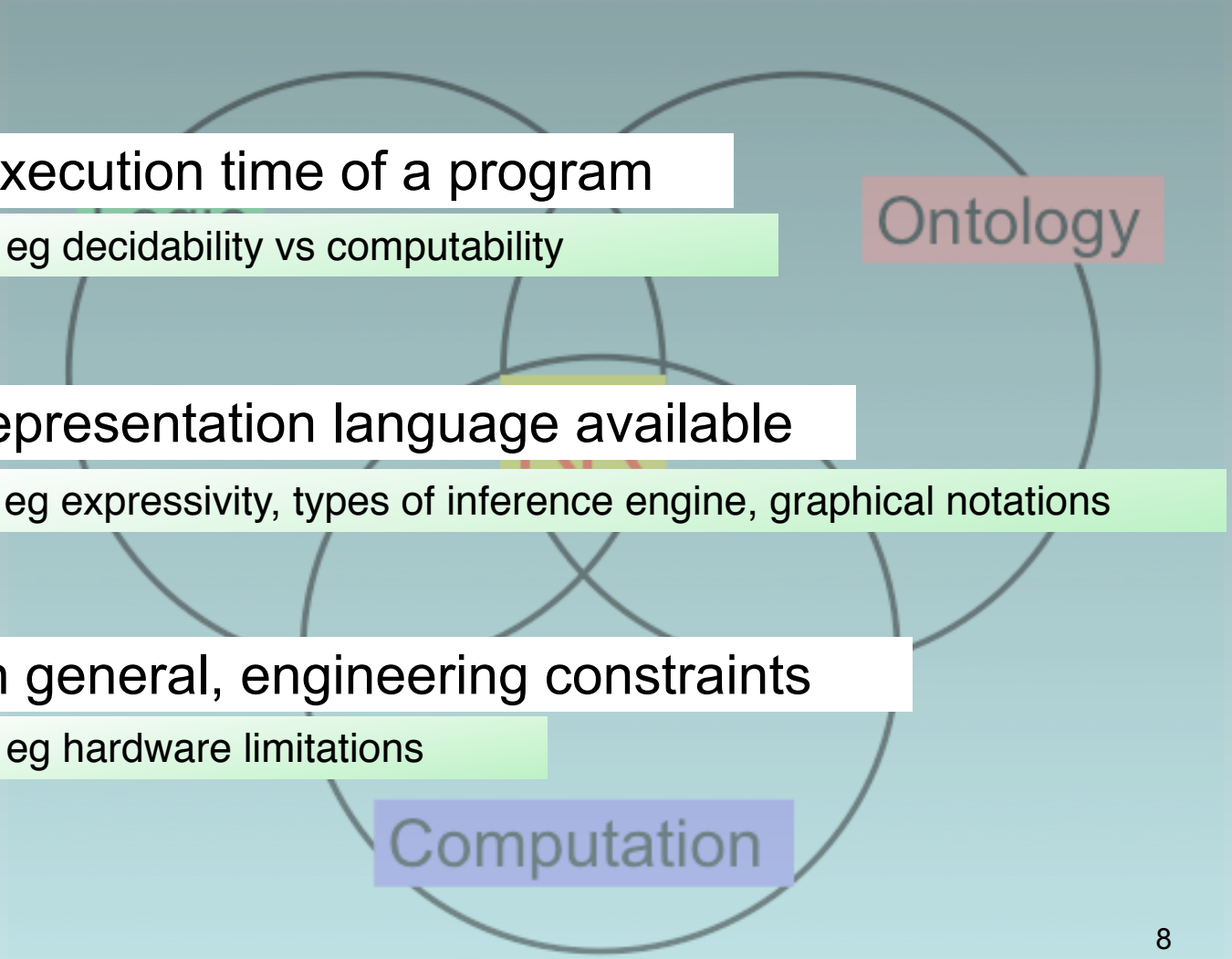
  eg decidability vs computability

- representation language available

  eg expressivity, types of inference engine, graphical notations

- in general, engineering constraints

  eg hardware limitations



John Sowa:

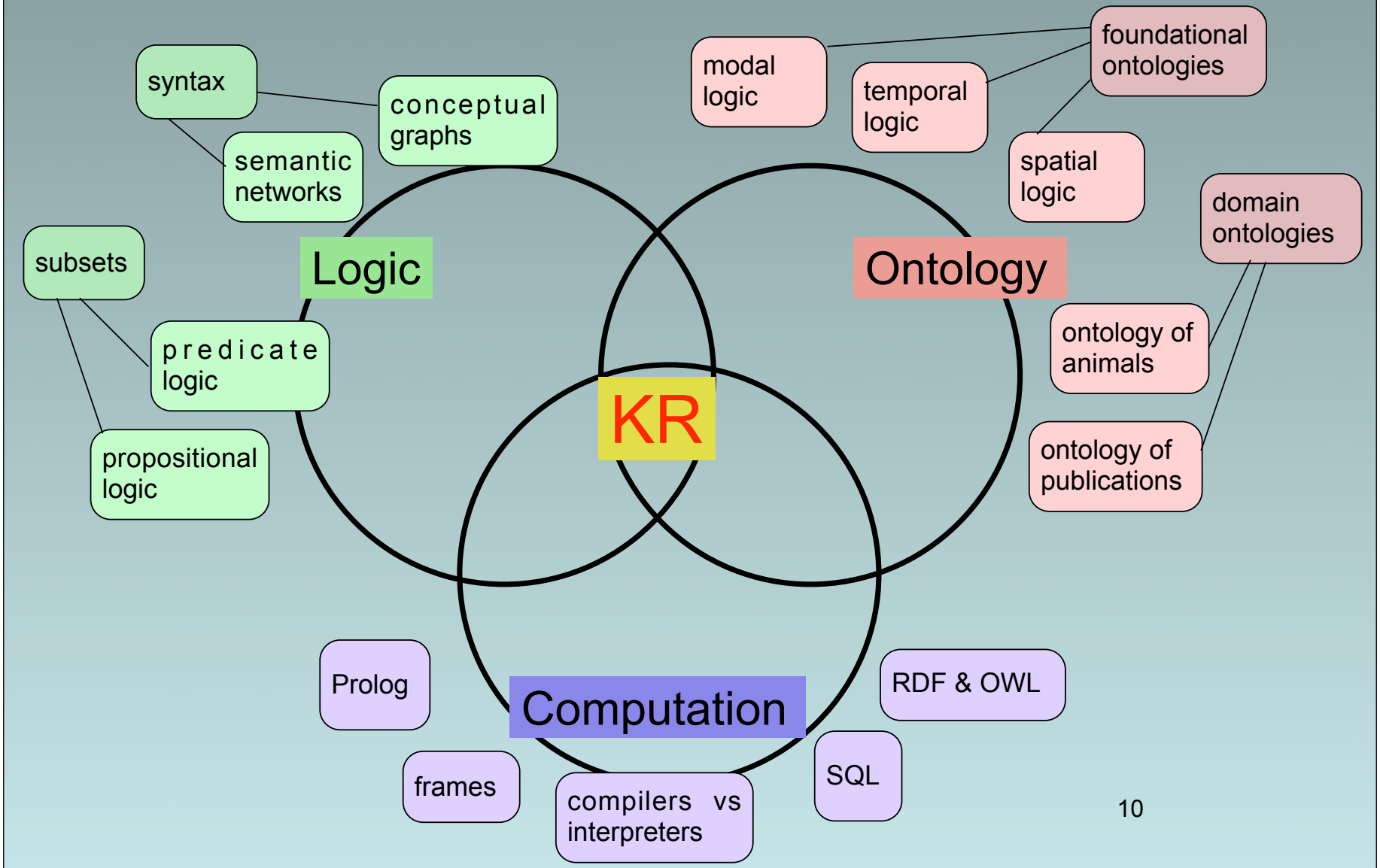
"**Without logic**, a knowledge representation is vague, with no criteria for determining whether statements are redundant or contradictory.

Without ontology, the terms and symbols are ill-defined, confused and confusing.

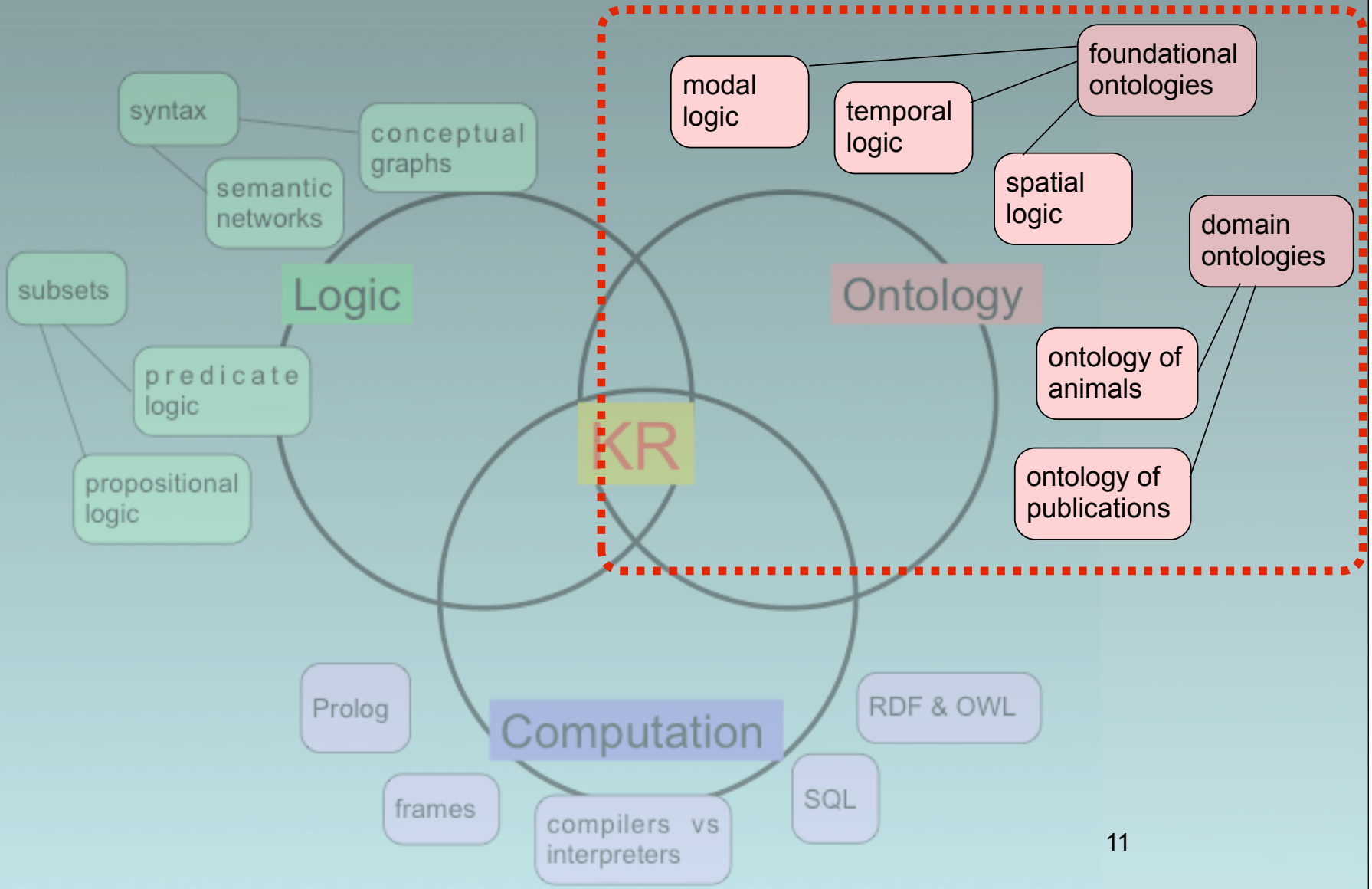
And **without computable models**, the logic and ontology cannot be implemented in computer programs.

Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain." (p. xii)

Possible research directions:



Possible research directions:



Pitfall [1]: Ontologies and data models

- main difference with data models is not the content, but the purpose

  - Clarity: context dependent vs context independent design

  - Extendibility: application oriented vs design for future reuse

  - Minimal Encoding Bias -avoid representational choice for benefit of implementation

- a conceptual model written in an ontology language is not necessarily an ontology!

  - you cannot see the difference by looking at the syntax

Pitfall [2]: Ontologies and knowledge bases

- the same languages (OWL, RDF-S, WSML, etc.) and the same tools and infrastructure can be used both for creating ontologies and for creating knowledge bases

  - not every OWL file is an ontology, since OWL files can also be used for representing a knowledge base (eg info about the concept of ‘city’, and the individual ‘Innsbruck’

- Ontologies are the vocabulary and the formal specification of the vocabulary only, which can be used for expressing a knowledge base

  - one initial motivation for ontologies was achieving interoperability between multiple knowledge bases!

Pitfall [3]: ontologies and XML Schemas

- XML schemas define a single representation syntax for a particular problem domain but not the semantics of domain elements.

e.g. sequence and hierarchical ordering of fields in a valid document instance, but do not specify the semantics of this ordering..

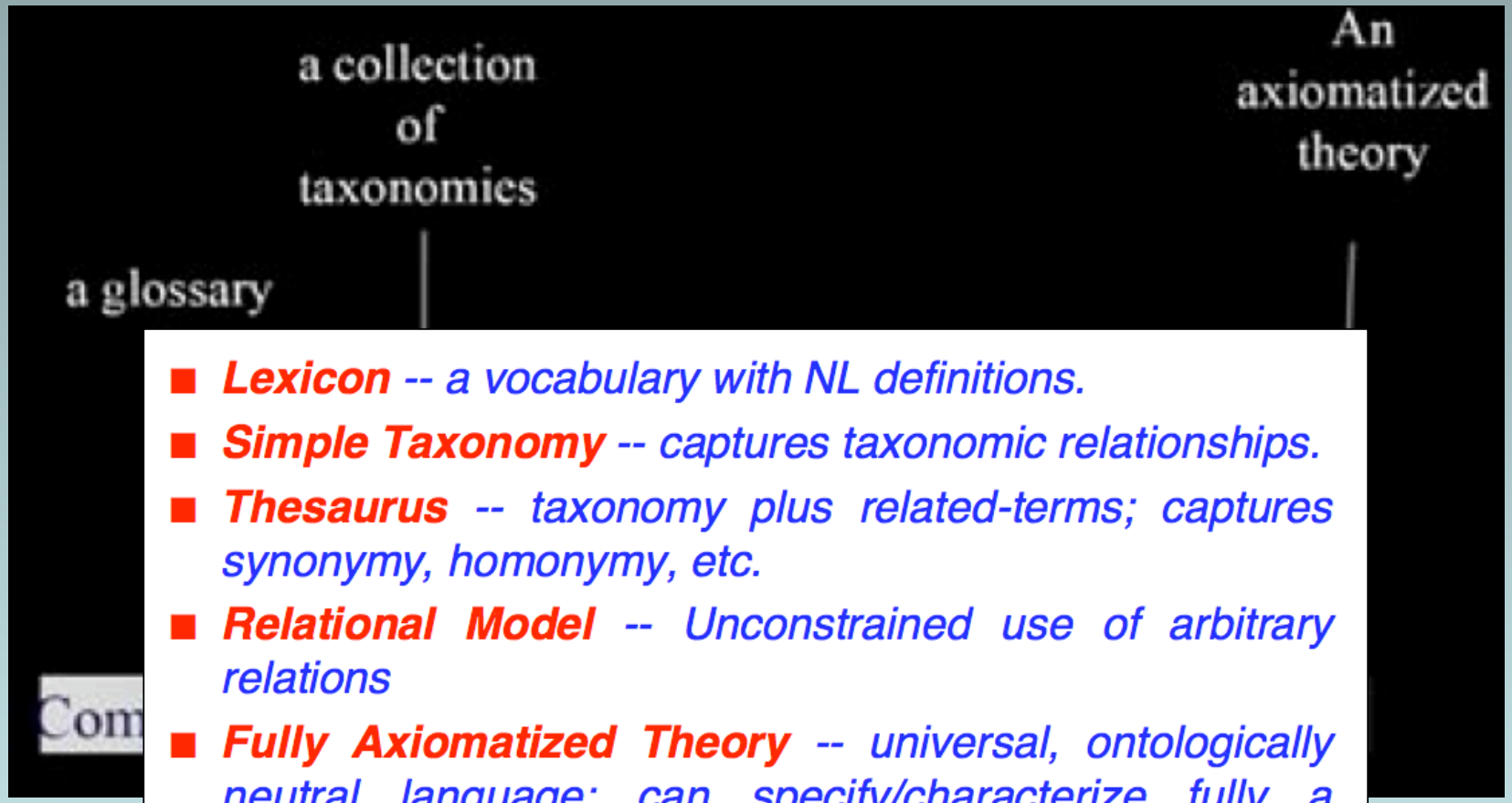
- They do not aim at carving out re-usable, context-independent categories of things

e.g. whether a data element “student” refers to the human being or the role of being as student.

- There is no standardized inference layer

To employ XML to generate new data, you need knowledge embedded in some procedural code somewhere, rather than explicitly stated, as in OWL.

Degrees of 'ontological depth'



- ***Lexicon*** *-- a vocabulary with NL definitions.*
- ***Simple Taxonomy*** *-- captures taxonomic relationships.*
- ***Thesaurus*** *-- taxonomy plus related-terms; captures synonymy, homonymy, etc.*
- ***Relational Model*** *-- Unconstrained use of arbitrary relations*
- ***Fully Axiomatized Theory*** *-- universal, ontologically neutral language; can specify/characterize fully a conceptualization.*

Upper vs Domain ontologies

- depends on the type of ‘predicates’ our (logical) theory is investigating..

  - domain **independent**: *part-whole, temporal relations, concrete-abstract, universal-particular, qualities*

  - domain **dependent**: *car makers, car materials, fuel consumption, etc.*

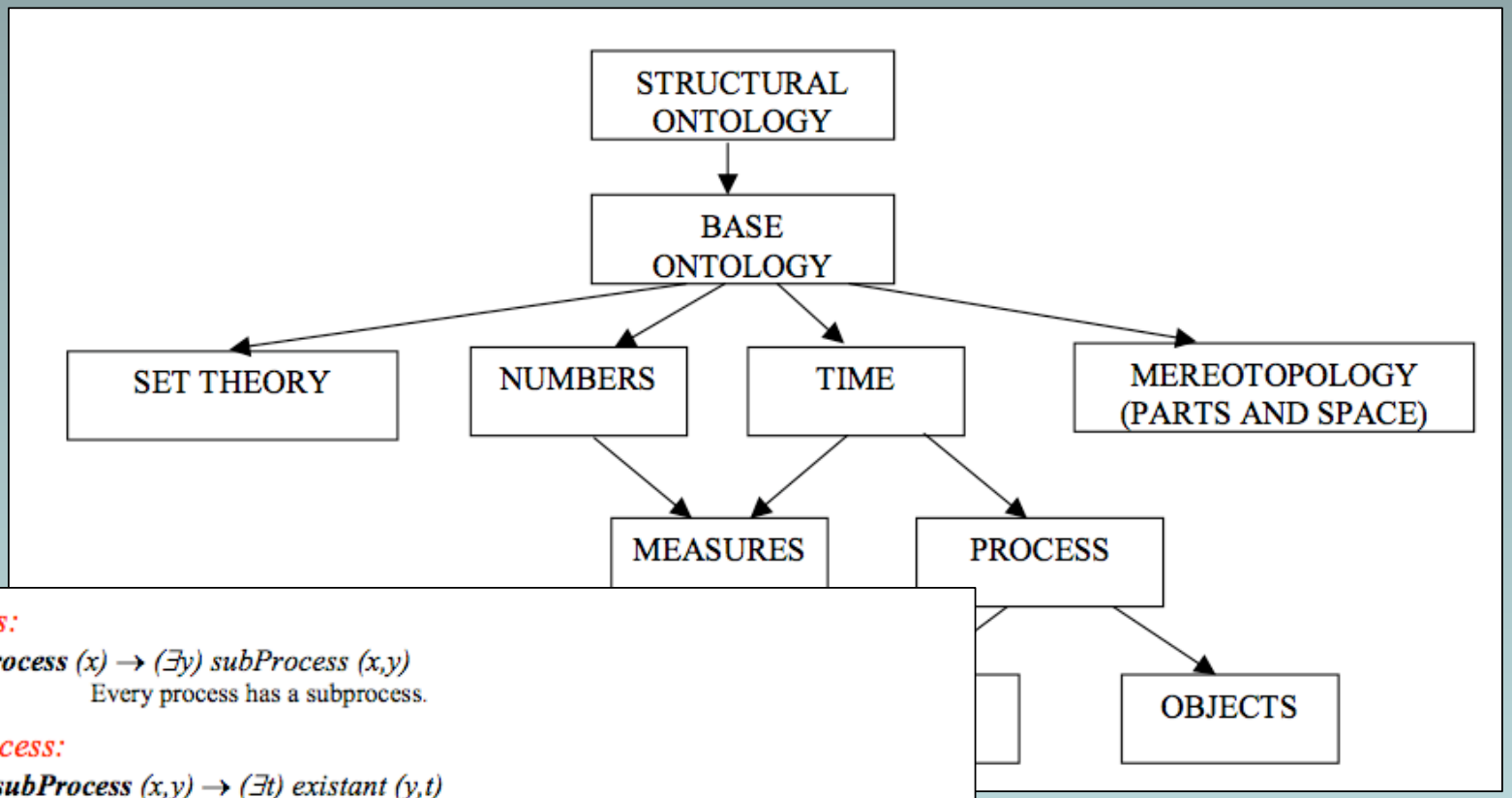
    - task ontologies: a problem solving process like diagnosis, monitoring, scheduling, design, and so on

- in the Semantic Web, top level ontologies are supposed to bridge the various possible domain ones

  - a top level ontology is very general and abstract

  - e.g. DOLCE, SUMO, CIDOC, CYC, BFO

E.g. top level of SUMO



Process:

(∀x) ***Process*** *(x) → (∃y) subProcess (x,y)*

Every process has a subprocess.

subProcess:

(∀x,y) ***subProcess*** *(x,y) → (∃t) existant (y,t)*

Every subprocess exists in a temporal instant

(∀x,y) ***subProcess*** *(x,y) → [(∀ z) located (y,z) → located (x,z)]*

A subprocess is located as the process

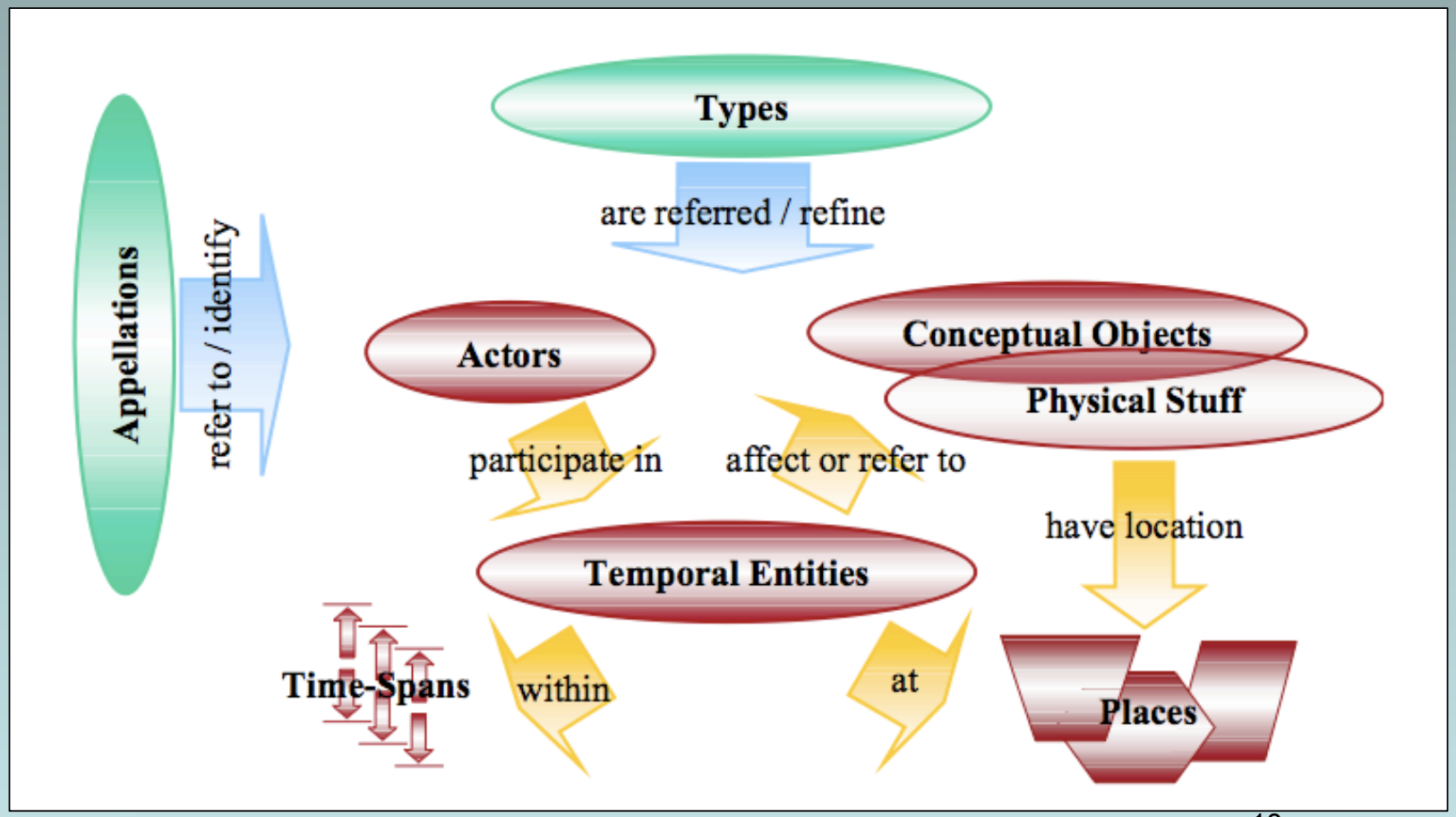
(∀x,y) ***subProcess*** *(x,y) → WhenFn(x)=WhenFn(y) ∨ during (WhenFn(x),WhenFn(y))*

Temporal inclusion of subprocesses

Niles and Pease. Towards a Standard Upper Ontology. FOIS'01 (2001)

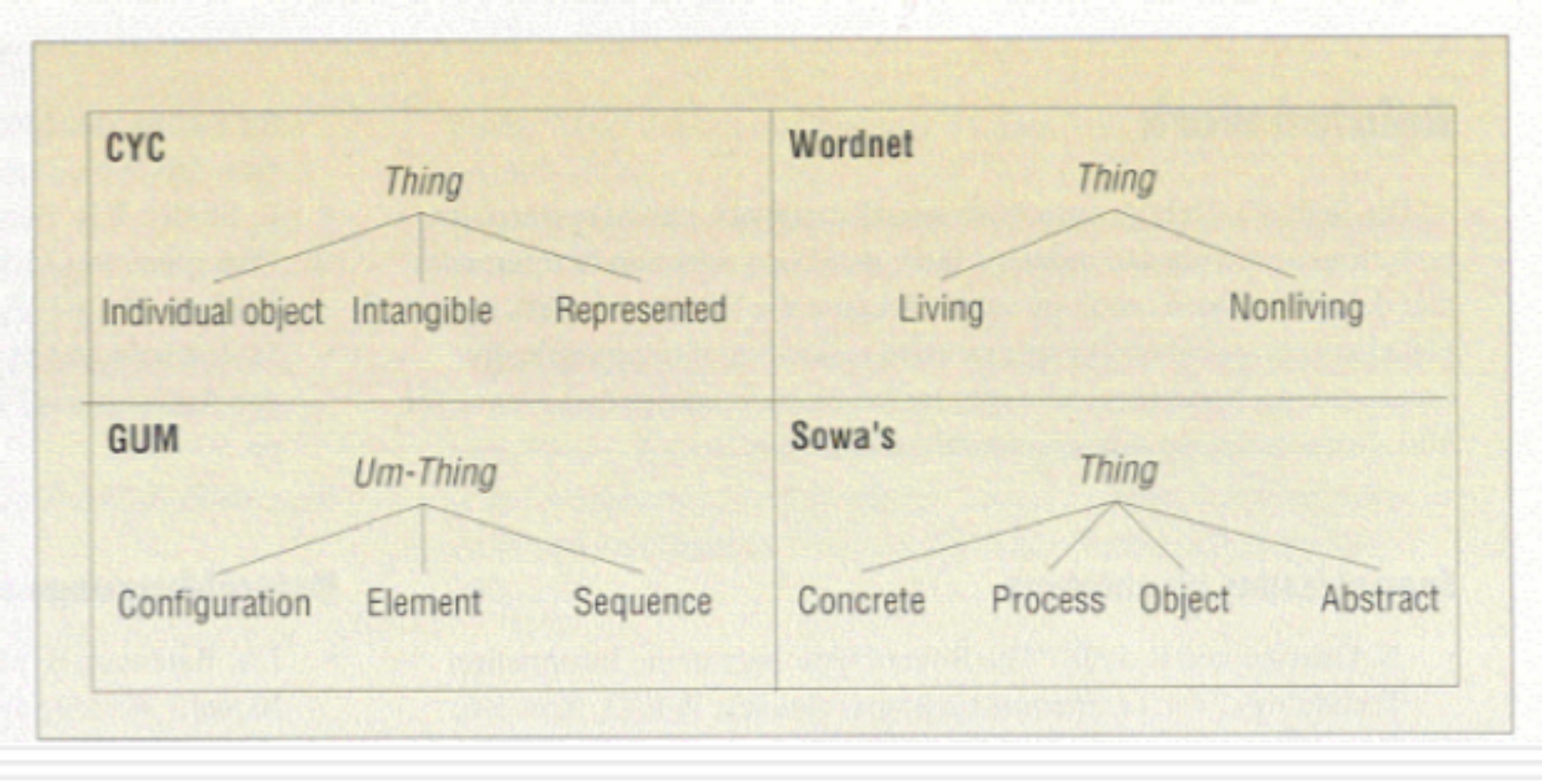
E.g. top level of CIDOC CRM

1996 ICOM initiative, 2006 **ISO standard** (version 4.2)

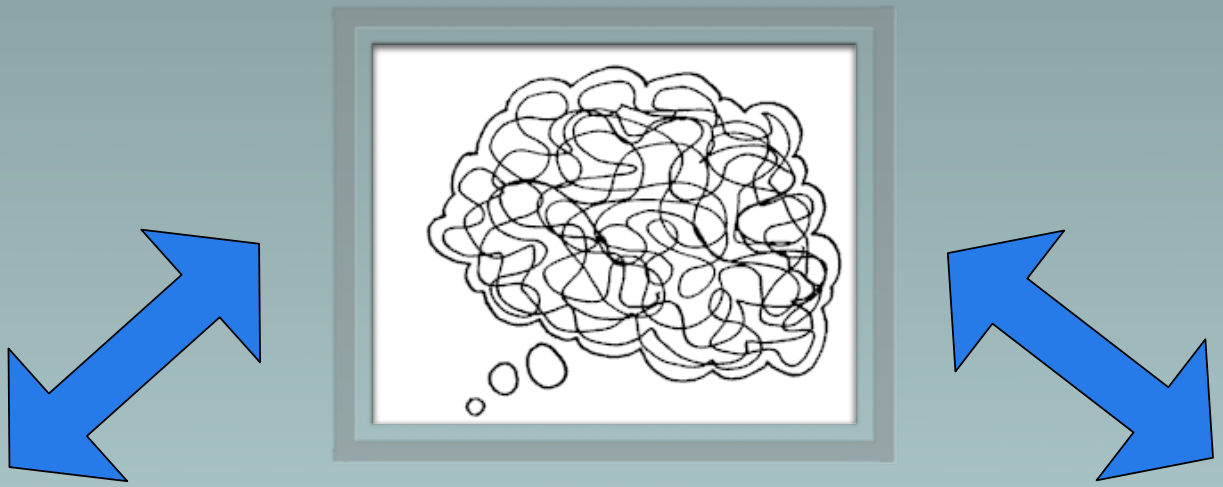


Doerr. The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. AI Magazine archive (2003) vol. 24 (3) pp. 75-92

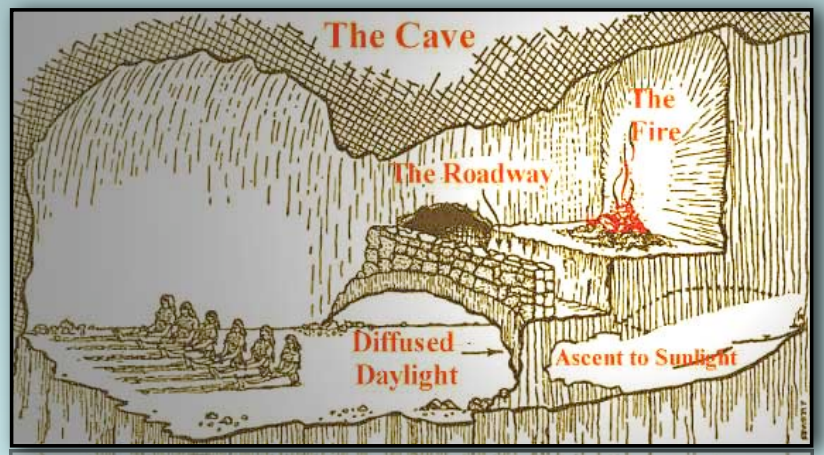
Upper ontologies: not only one proposal!



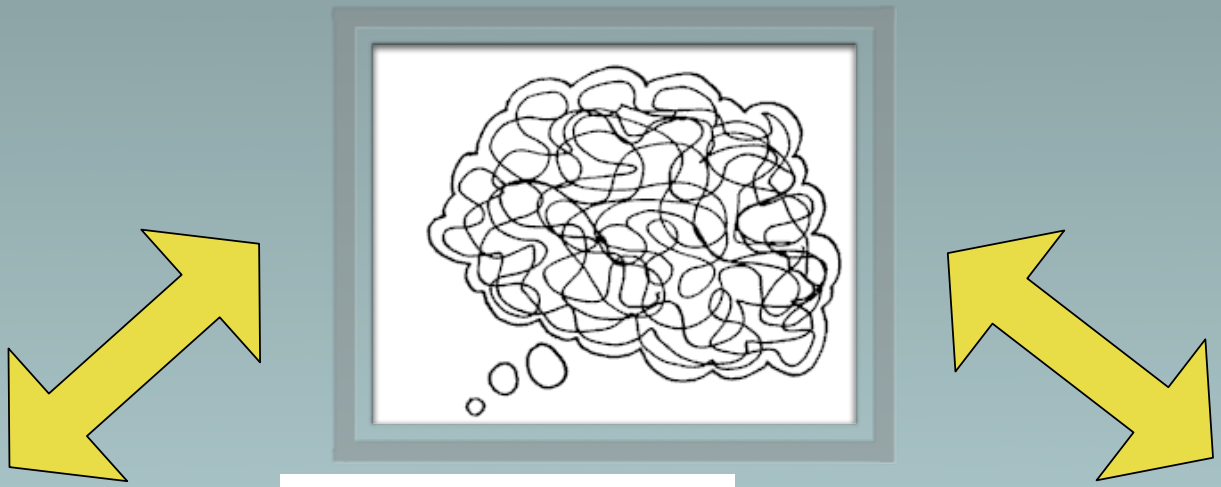
‘Realist’ vs ‘Conceptualist’ ontologies:





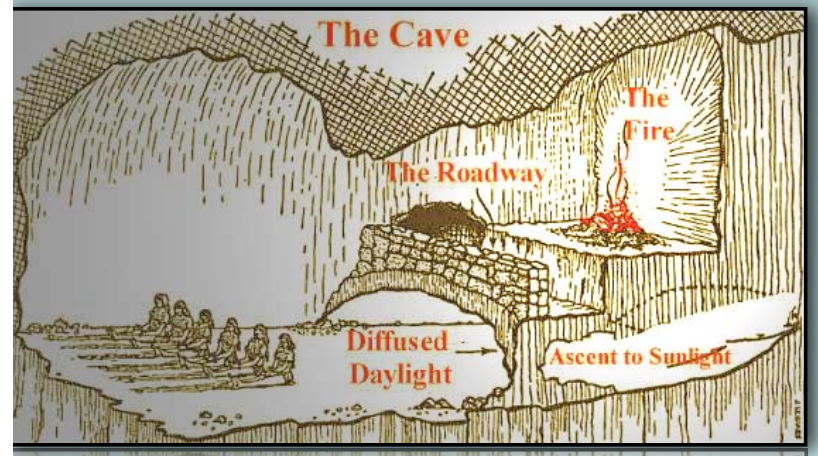


‘Realist’ vs ‘Conceptualist’ ontologies:





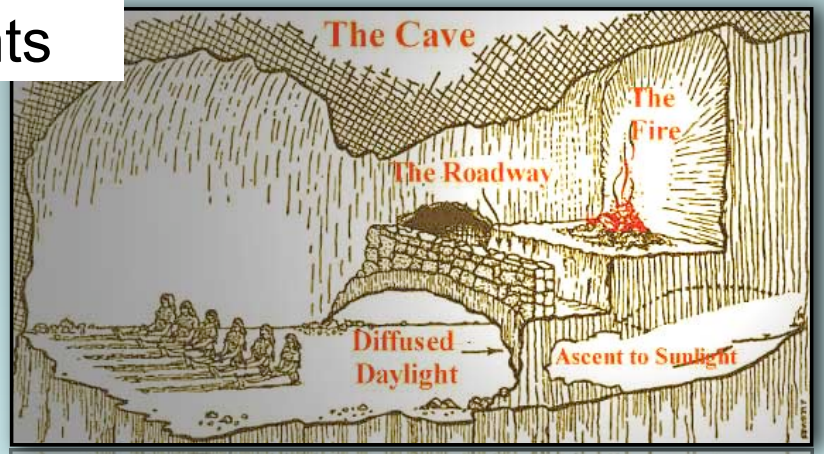
eg DOLCE: reality is socially constructed; ontologies should have a ‘cognitive bias’



‘Realist’ vs ‘Conceptualist’ ontologies:

eg BFO: ontologies mirror the ‘true’ reality, that is what is discovered by the latest scientific experiments



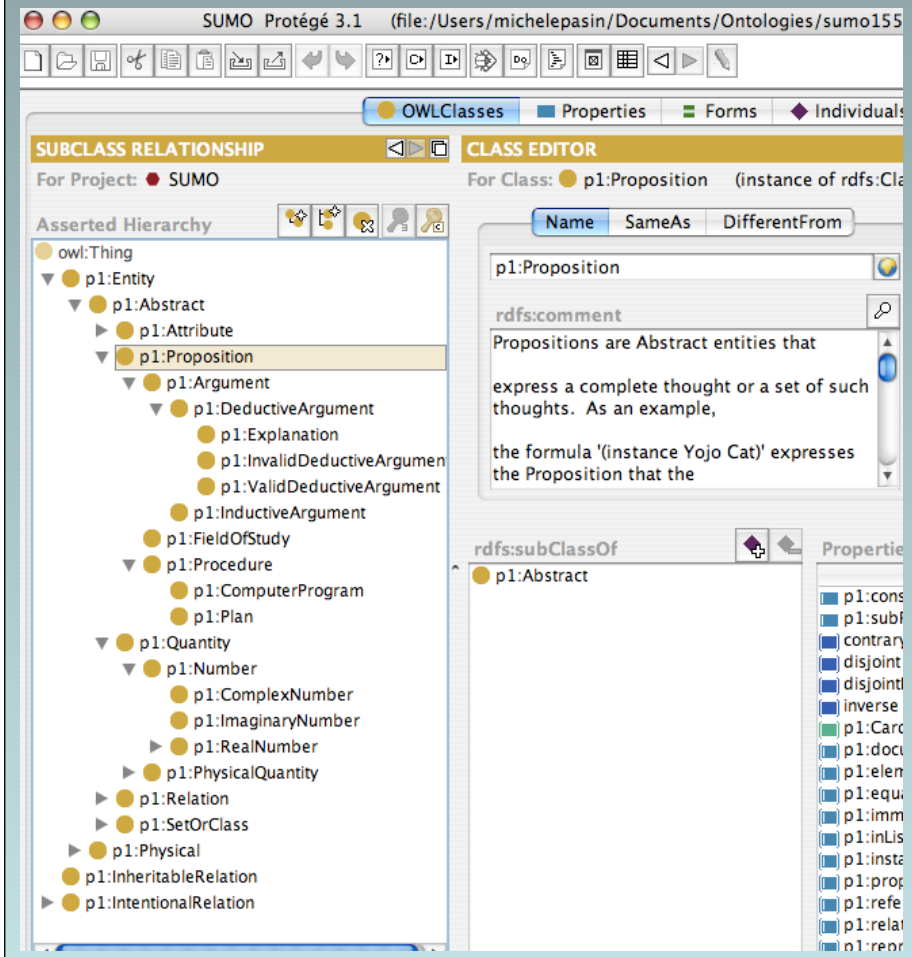


what is it good for?

What is an ontology (as KR) good for?

- to enable **data exchange** among programs
- to simplify unification (or **translation**) of disparate representations
- to employ **knowledge-based services**
- to embody the representation of a **theory**
- as a **reference** to guide new formalizations
- to facilitate **communication** among people
- to **find** or **browse** data
- to **reason** with data when you find it
- to **label** the data you are collecting
- to build a knowledge model that will **stand the test of time**

Principle #1: ontology as a program



1. An ontology is an explicit, formal specification of a **theory**

2. An ontology is a **model** that can be manipulated by a computer

3. An ontology can be *run* as we run computer **programs**

Principle #2: ontology as a contract

A Pragmatic Perspective

- Ontologies are not about truth or beauty.
- They are agreements, made in a social context, to accomplish some objectives.
- It's important to understand those objectives, and be guided by them.



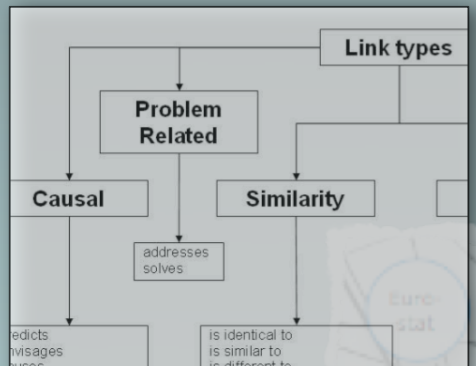
Gruber. It Is What It Does: The Pragmatics of Ontology. Invited presentation to the meeting of the CIDOC Conceptual Reference Model committee (2003)

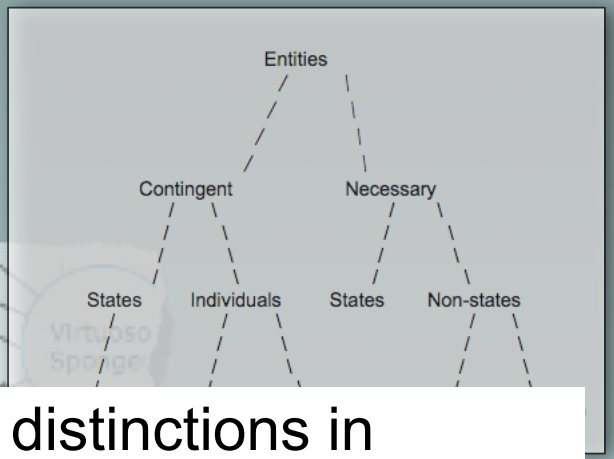
software applications

research communities

how do we build good ontologies?

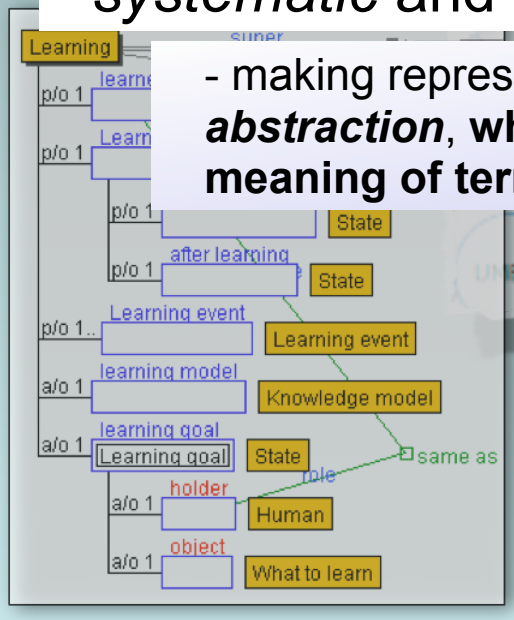
Reusing philosophical methods&notions in KR

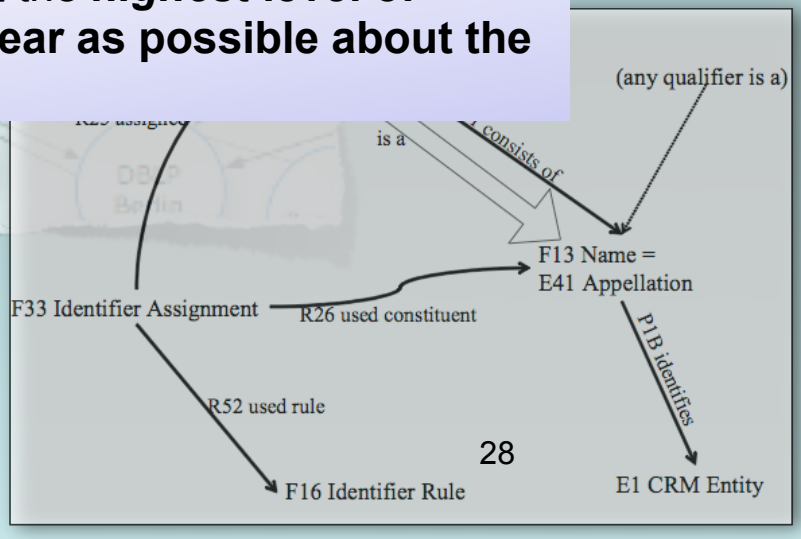




- a *theory* of how to make ontological distinctions in *systematic* and *coherent* manner

- making representational choices at the ***highest*** **level of *abstraction*, while still being as clear as possible about the meaning of terms**





A few generic principles…

- determine an **essential** property for each concept and instance

  - Proper use of is-a relation should inherit the “Essential” property of its super classes (= identity criteria checking)

- **concepts** rather than **terms**

  - people are easily trapped by the endless terminological discussion departing from the underlying conceptual structure of the target domain

- **role** concepts vs **basic** concepts

  - Clear and consistent differentiation between basic concepts (man, rice, oil, etc.) and role concepts(teacher, food, fuel, etc.).

The ‘ontoclean’ methodology (Guarino, Welty)

Guarino and Welty. Evaluating ontological decisions with OntoClean. Commun. ACM (2002) vol. 45 (2) pp. 61-65

- **4 fundamental metaproperties of concepts/properties**
  - *Identity: the possibility to distinguish two instances of a concept in base of characteristic property*
    Eg. ‘Human: ‘having same fingerprint’
    ‘Species’: having a place in a biological taxonomy
    (indeed Human is a Species but it is not subsumed by Species)
  - *Dependence: Property P depends on property Q, if, when Q is true, P is true too*
    Eg. ‘Student’ depends on ‘School’
  - *Rigidity: if a property is essential for all instances*
    Eg. ‘Person’ is rigid; ‘Student’ is not rigid
    (indeed a person is not a student, a person plays the role of a student)
  - *Unity: all parts identified with a unifying relation*
    Eg. ‘ocean’ is a whole, ‘water’ is not
    (indeed, an ocean is constituted by water, ocean is not water)

slide adapted from Boella. Ontologies and the Semantic Web. Scienze Cognitive 2002-2003 course (2002)

Why metaproperties?

- **Constraints on the subsumption relation**

1. **A non rigid class (-R) cannot subsume a rigid one (+R)**
   Ex. 'Student' cannot subsume 'Person'

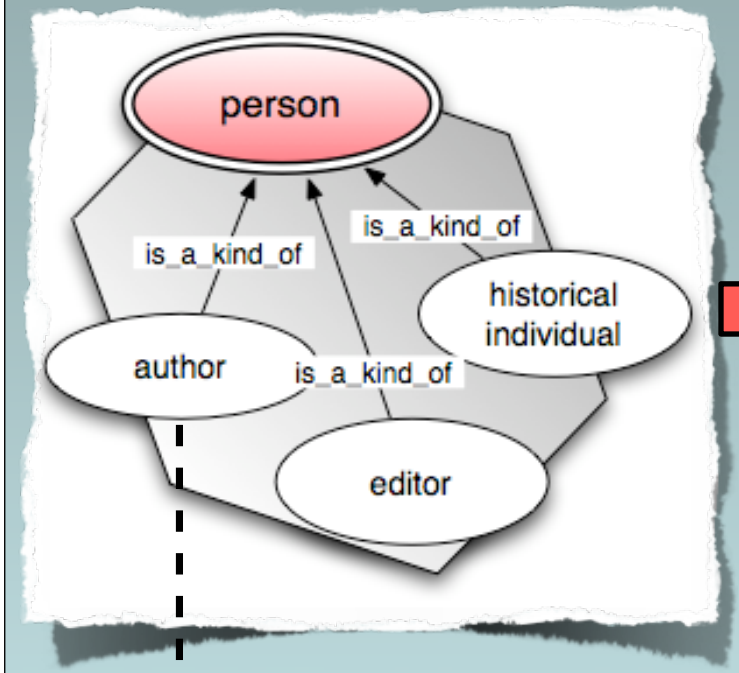
2. **A class with identity (+I) cannot subsume one without (-I)**
   Ex. 'Park' cannot subsume 'Location'

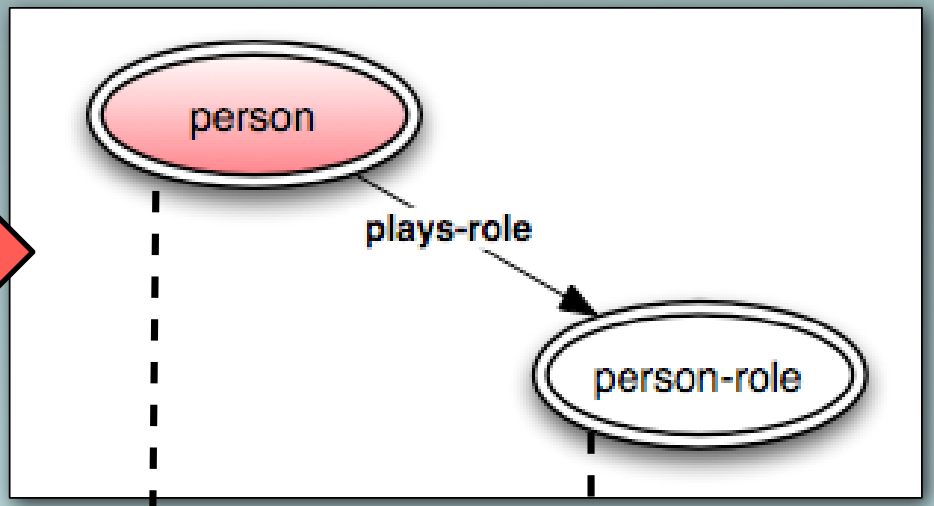
3. **A class with unity (+U) cannot subsume one without (-U)**
   Ex. 'Amount of Matter' cannot subsume 'Physical Object': if some amount is removed from a quantity the quantity is not itself anymore. A person remains the same even if hair is cut

4. **Dependent properties (+D) cannot subsume independent ones (-D)**
   Ex. 'Student' cannot subsume 'Person'

slide adapted from Boella. Ontologies and the Semantic Web. Scienze Cognitive 2002-2003 course (2002)

Example: looking for essential properties... #1



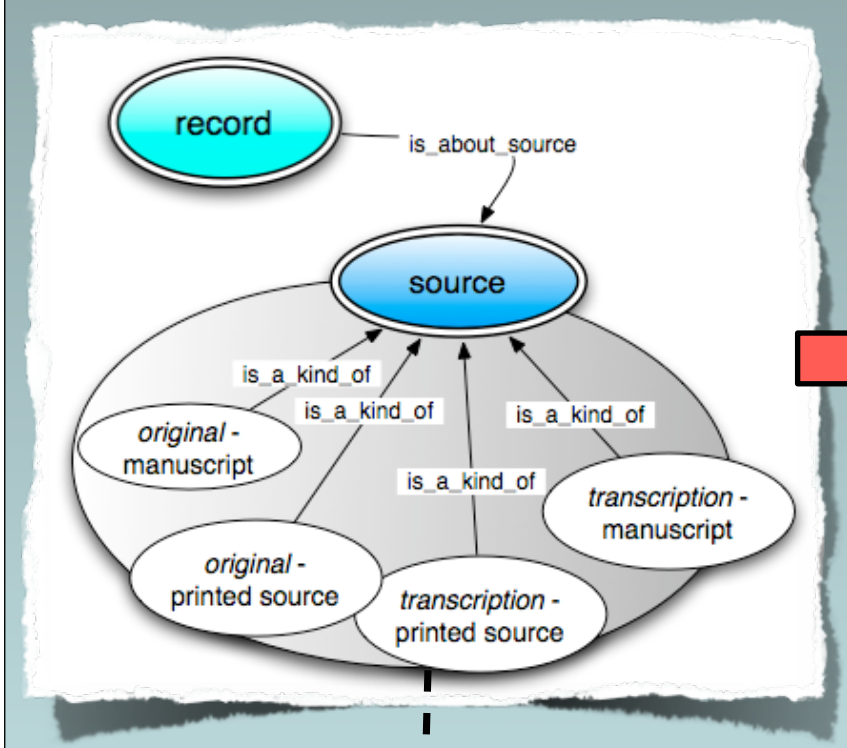


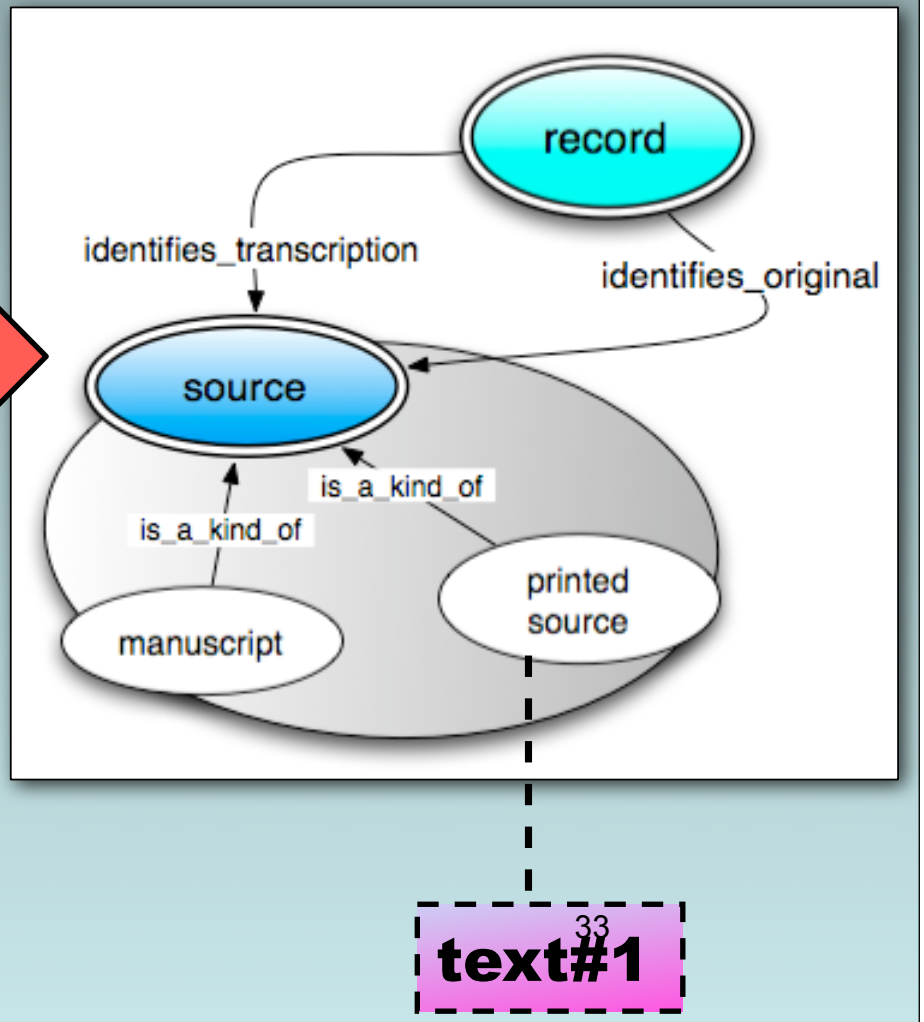
Mr. Jones

Mr. Jones

author, editor, common person...

Example: looking for essential properties... #2





Common ‘things’ we mention in our contracts:

- information objects
  - key characteristics of entities that can carry information, that can be seen as (or part of) a *representation*
- physical features of information objects
  - e.g., materials, conditions, preservation ...
- abstract features of information objects
  - e.g., the *contents* of an information object, the Hamlet as a work
  - e.g., the *linguistic* features of an information object (latin, english, etc.)
  - e.g., aspects of the *discourse* used to communicate the contents of an information object (e.g., proem, dispositive word, bound, curse etc.). These aspects will vary with different projects!

Conclusion: ontologies at CCH ?

- what for?

- shall we work on specific domains...
  - or need a foundational one ?

- lots of stuff for next sessions
  - domain ontologies
  - implementation languages
  - storage layers