Laboratory for Applied Ontology

Institute of Cognitive Science and Technology
Italian National Research Council

Ontological Analysis

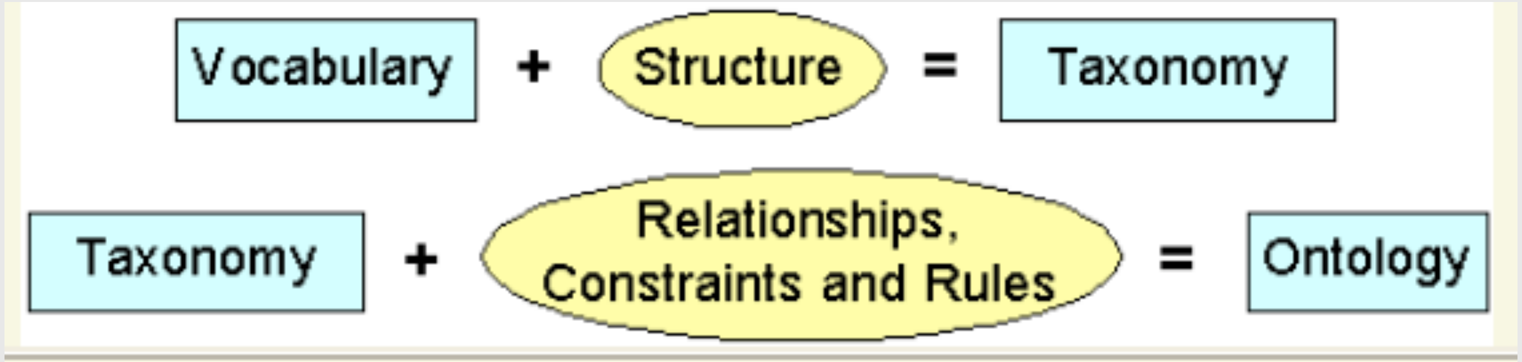
3 - Ontology-driven information systems

Nicola Guarino
Laboratorio di Ontologia Applicata (LOA)
Istituto di Scienze e Tecnologie della Cognizione (ISTC-CNR)
Trento, Italy

When is a precise (and well-founded) ontology useful?

1. When *subtle distinctions* are important
2. When *recognizing disagreement* is important
3. When *general abstractions* are important
4. When *careful explanation and justification* of ontological commitment is important
5. When ***mutual understanding*** is more important than interoperability.

Ontologies and taxonomies



Ontologies vs. classifications

- Classifications focus on:
  - ***access***, based on pre-determined criteria (encoded by ***syntactic keys***)

- Ontologies focus on:
  - ***Meaning*** of terms
  - ***Nature*** and ***structure*** of a domain

Ontologies vs. Database Schemas

- Database schemas:
  - Constraints focus on ***data integrity***
  - Relationships and attribute values out of the DoD
  - Typically ***non-executable***

- Ontologies:
  - Constraints focus on ***intended meaning***
  - Relationships and attribute values first class citizens
  - Typically ***executable***

Ontologies vs. Knowledge Bases

- Knowledge base
  - Assertional component
    - **reflects *specific (epistemic) states of affairs***
    - designed for ***problem-solving***
  - Terminological component (*ontology*)
    - ***independent*** of particular ***states of affairs***
    - Designed to support ***terminological services***

Ontological formulas are (assumed to be)
invariant, necessary information

Ontology-driven information systems

Ontology-Driven Information Systems

- Every IS ***has*** its own ontology (either implicit or explicit)
- The ODIS perspective: ***explicit*** ontologies play a ***central*** role, driving ***all*** aspects and components of an IS

- Two (main) dimensions to assess the role of an explicit ontology:
  - ***temporal dimension***: development time vs. run time
  - ***structural dimension:*** impact on the various IS components:
    - ***database component***
    - ***application program***
    - ***user interface***

Temporal dimension: *development time*

- Two scenarios:
  - A pre-existing ***ontology library*** containing domain and task ontologies as “main building blocks” to be adapted and rused
    - standard IS: the ontology content is *embedded* in the standard components
    - ODIS: an *application ontology* is built by specializing domain and task ontologies taken from the library
  - Only an ***upper-level ontology*** available: not building blocks, but ***conceptual tools*** (analogous to other CASE tools)

- Two kinds of development:
  - IS ***engineering***
  - IS ***re-engineering***

Temporal dimension: *run time*

- Ontology-***aware*** IS: the IS just uses the ontology for some specific purpose
- Ontology-***driven*** IS: the ontology is a ***central componen***t of the IS, cooperating at run time towards its “higher” overall goal

- Important application: ***inter-agent communication***

Structural dimension: *the database component*

- Development time:
  - support to ***requirement analysis and conceptual modelling*** (integrated with lexical resources like WordNet)
  - development of a ***global conceptual schema*** (DB integration)
- Run time:
  - mediation-based approach to ***information integration***
  - ***intensional queries***

Structural dimension: *the user-interface component*

- Development time:
  - Generation of ***form-based interfaces*** (constraints checking)
- Run time:
  - Support ***quering and browsing the ontology*** itself:
    - better understanding of the vocabulary
    - queries at the desired level of specificity
  - ***Vocabulary detaching***:
    - user free to adopt his own NL terms (mapped - after disambiguation - to the IS vocabulary with the help of the ontology)

Structural dimension: *the application program component*

- Development time:
  - Generation of the static part of a program (type structure)
  - Support to OO design

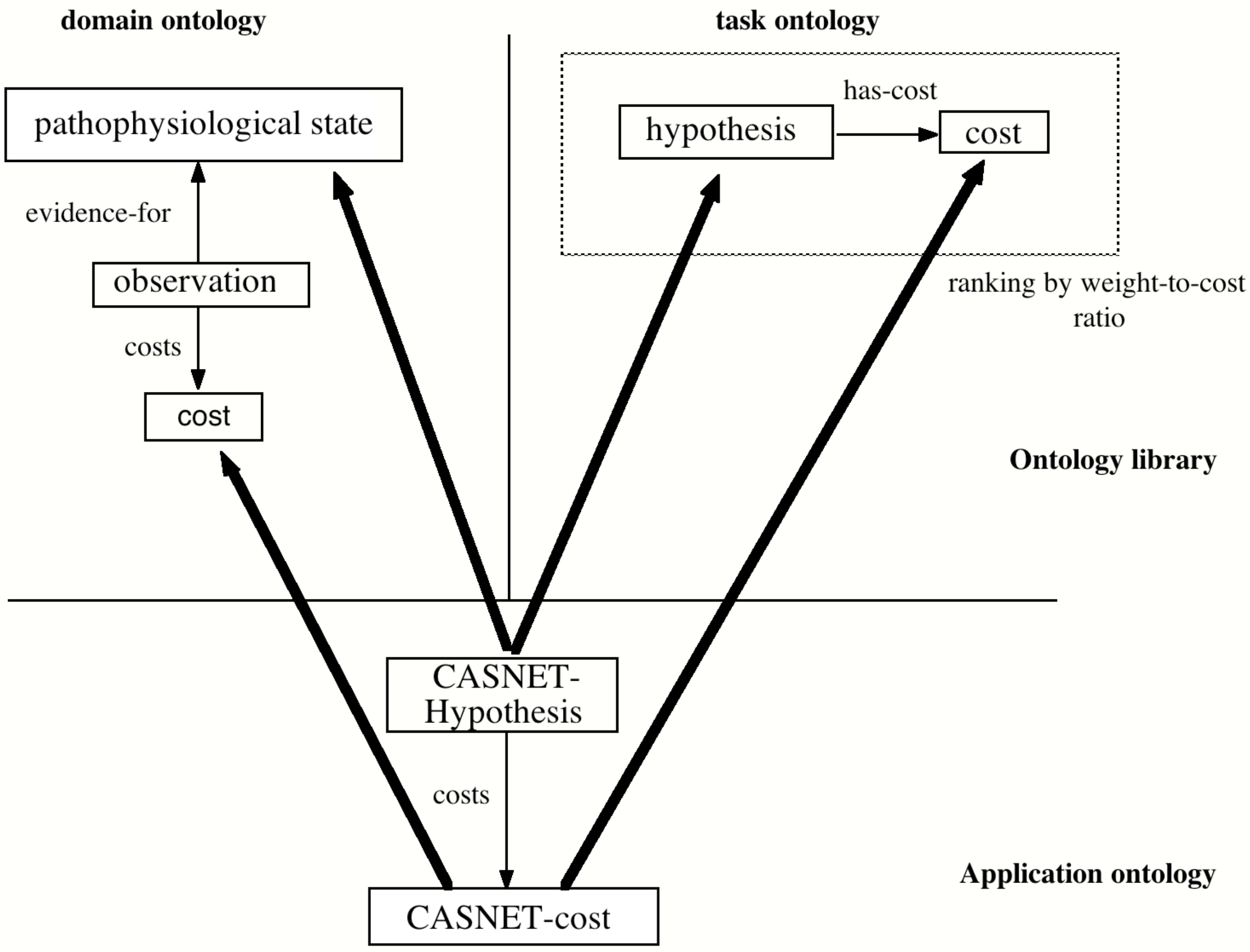
- Run time:
  - Explicit account of the ***ontological commitment*** of an application program
  - Increase of the ***transparency*** of application software

The task dependency problem

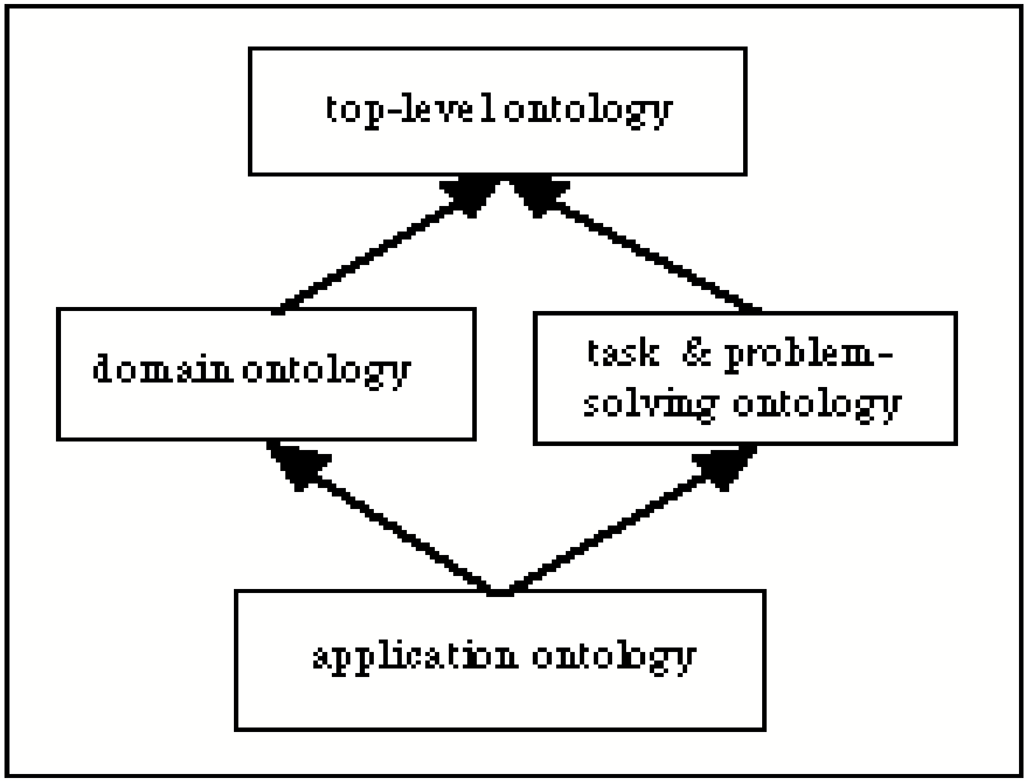
Representing knowledge for the purpose of solving some problem is strongly affected by the nature of the problem and the inference strategy to be applied to the problem.

[Bylander & Chandrasekaran 1988]

Making task dependence explicit [Guarino 97]



The role of task* ontologies



Without explicit domain and task ontologies, semantic interoperability among application ontologies is a myth!

(*) *substitute "task" with "service" if you want to be trendy…*

A single, imperialistic ontology?

- An ontology is first of all ***for understanding each other***
  - ...among people, first of all!
  - not necessarily for thinking in the same way

- A single ontology for multiple applications ***is not necessary***
  - Different applications using different ontologies can co-exist and co-operate (not necessarily inter-operate)
  - ...if linked (and compared) together **by means of a general enough basic categories and relations (*primitives*)**.

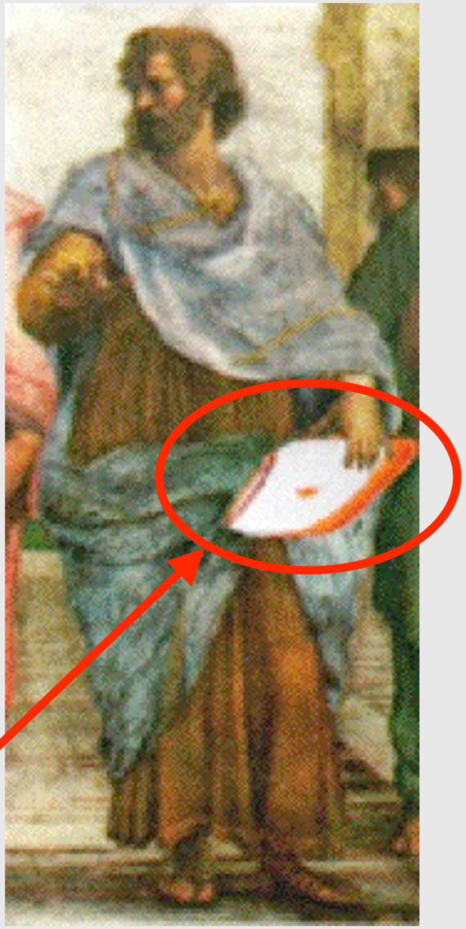
- If basic assumptions are not made explicit, any imposed, common ontology risks to be
  - seriously mis-used or misunderstood
  - opaque with respect to other ontologies

Which primitives?
The role of ontological analysis

- Theory of Essence and Identity
- Theory of Parts (Mereology)
- Theory of Wholes
- Theory of Dependence
- Theory of Composition and Constitution
- Theory of Properties and Qualities

The basis for a common ontology vocabulary

Idea of Chris Welty, IBM Watson Research Centre, while visiting our lab in 2000



The problem of primitives

- Representation primitives vs. ***ontological primitives*** (against ***arbitrary*** interpretations)

- Let's aim at ***general*** primitives, similarly to what happens in mathematics: ***set, relation, transitive, symmetric…***

The Ontological Level

(Guarino 94**)**

Level	*Primitives*	*Interpretation*	*Main feature*
Logical	Predicates, functions	Arbitrary	Formalization
Epistemological	Structuring relations	Arbitrary	Structure
Ontological	**Ontological relations**	**Constrained** (meaning postulate s)	**Meaning**
Conceptual	Conceptual relations	Subjective	Conceptualization
Linguistic	Linguistic terms	Subjective	Language dependence

The semantic web architecture [Tim Berners Lee 2000]