Ontology Summit 2013:
Ontology Evaluation Across the Ontology Lifecycle
Virtual Panel Session 09, Track C – March 14, 2013

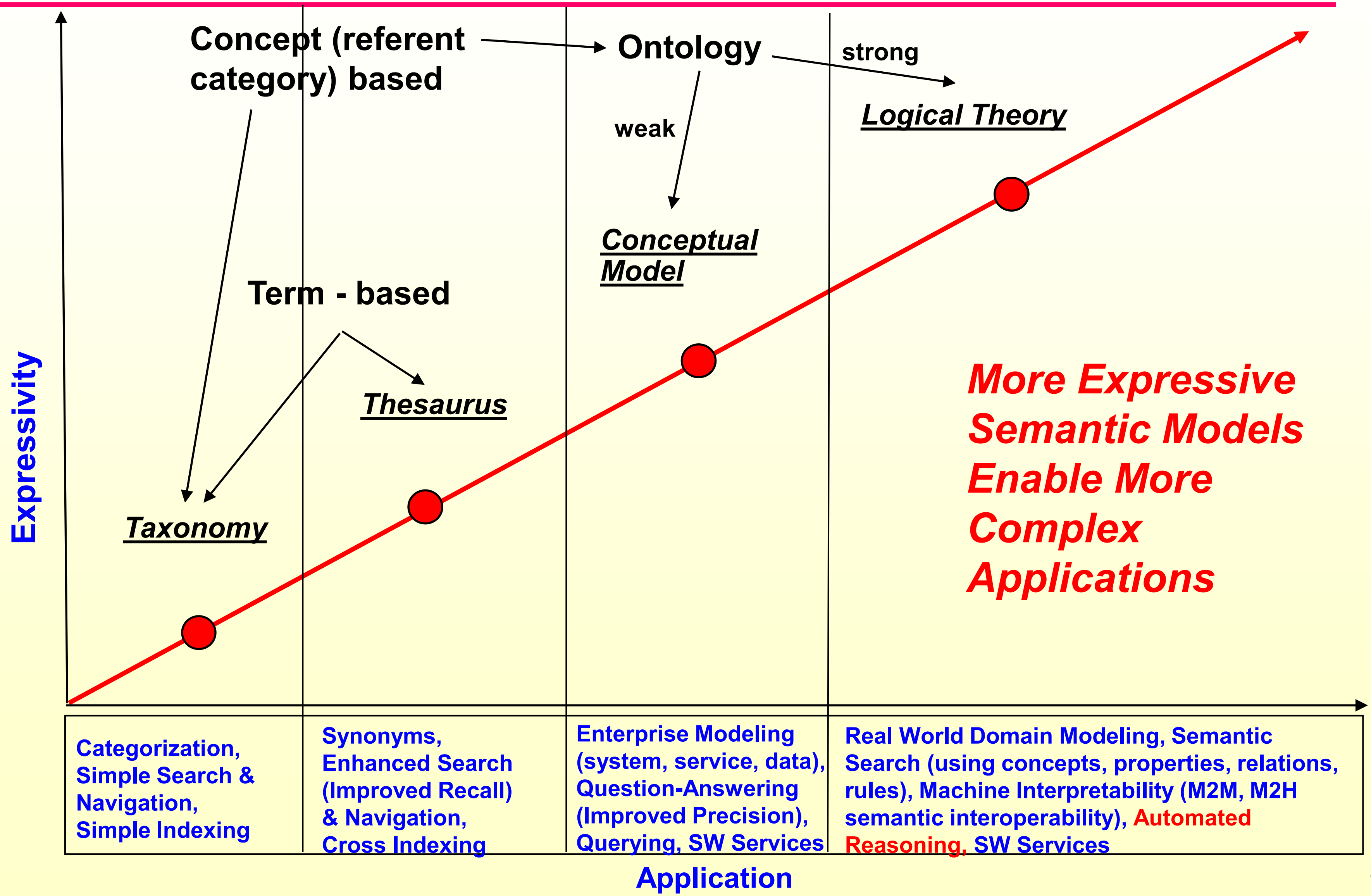
Developing Quality Ontologies Used for Reasoning

Dr. Leo Obrst
Information Semantics
Information Discovery & Understanding
Command & Control Center
MITRE

General Methodological Issues

- Assume every new ontology will be developed for automated reasoning unless explicitly ruled-out (and knowing the consequences of ruling it out)
  - This means: know the rough complexity of the semantic model you need
  - E.g., if your reasoning requirements are very large-grained (e.g., determining which topic bucket should this document be placed in) you probably don't need an ontology and the reasoning you need is minimal
- Typically you will want to reason over both the classes and the instances, i.e., what kinds of things are there, and what are the things of that kind?
  - Description Logics: T-Box vs. A-Box; but most ontology languages do not make such a hard/fast distinction
- Choose an ontology reasoning architecture: depends on the kind of reasoning you will do
  - DL classificational reasoning only?
  - Real "rule" reasoning?

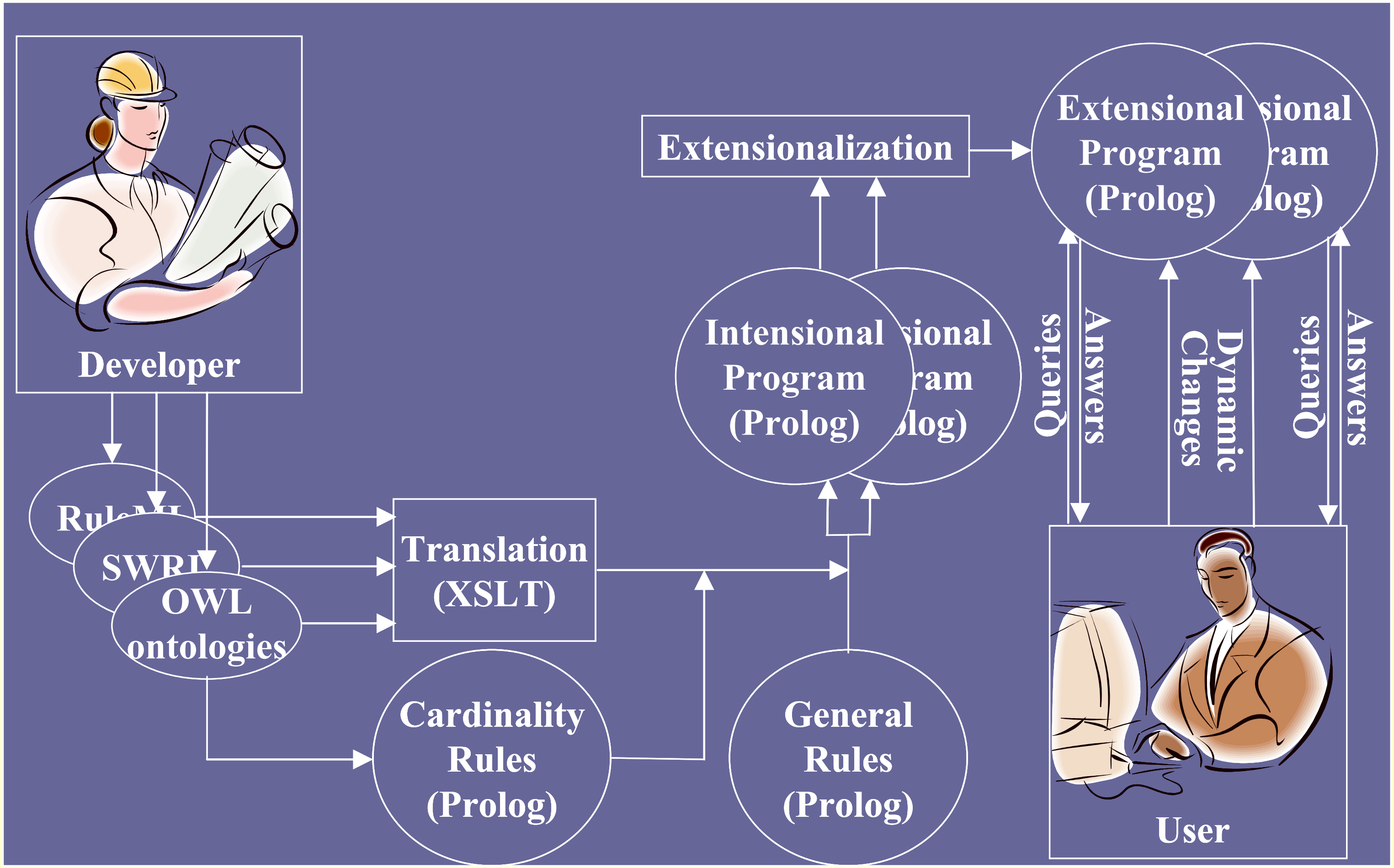
Ontology Spectrum: Complexity of Applications



What do you want to do?
What kind of reasoning?

- Build an ontology, build a knowledge base
- Check consistency of your knowledge
- Check completeness of your knowledge
- I.e., Model checking, model finding
- Automatically classify new concepts, assertions
- Query the KB (search & navigation)
- Perform other inference (sometimes called rule-based reasoning)
  - Deduction
  - Induction
  - Abduction
- Add probabilistic reasoning
- Reason over beliefs (Truth Maintenance Systems), i.e., evidential reasoning
- Have built in modal operators: necessity/possibility, obligation/permission/prohibition, temporal, etc.
- **No Reasoning without Representation!**

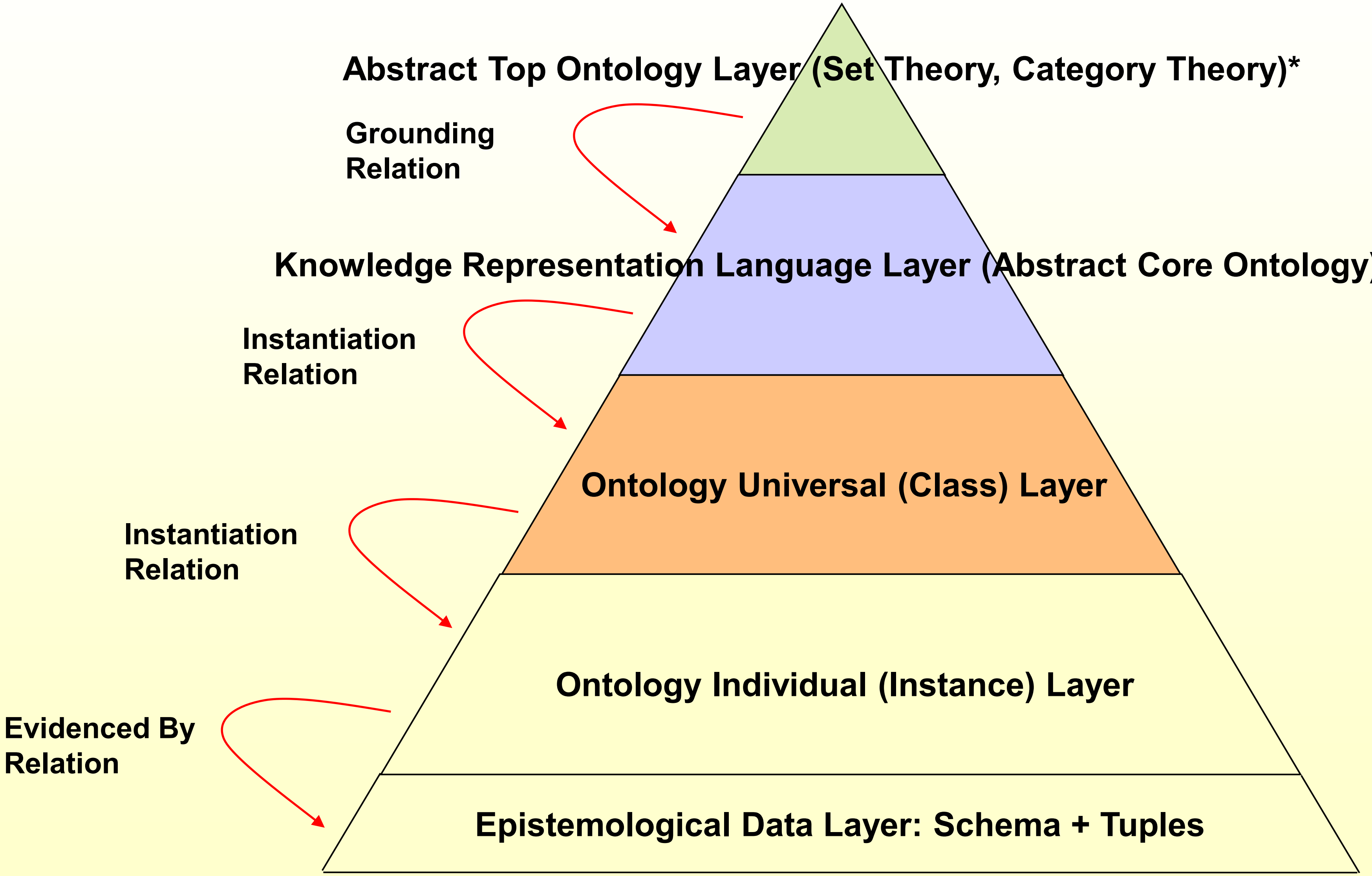
2004-2007: We had to develop our own ontology/rule reasoning system



Rules not expressible in SWRL were represented in Prolog directly

Ontology Content Architecture: You Need an Architecture!

* Adapted from: Herre, Heinrich, and Frank Loebe. 2005. A Meta-ontological Architecture for Foundational Ontologies. In: R. Meersman and Z. Tari (Eds.): CoopIS/DOA/ODBASE 2005, LNCS 3761, pp. 1398–1415, 2005. Springer-Verlag Berlin Heidelberg.



Ontology Evaluation Criteria

- Ontology coverage of a particular domain
  - The richness, complexity and granularity of that coverage
  - The specific use cases, scenarios, requirements, applications, and data sources it was developed to address
- Formal properties of the language in which it is modeled:
  - ***Soundness*****:** any expression that can be derived from the knowledge base (KB) of the ontology and its instances is logically implied by that KB)
  - **C*ompleteness*:** any expression that is logically implied by the KB can be derived
  - ***Decidability***:being both sound and complete). All of these will correlate with the formal *complexity* (time of execution, space of memory needed to compute an answer. Decidability of a language or logic does not mean tractability of the automated reasoning on that language, but there is a relationship
  - ***Consistency:*** can contradictions be proven?
  - E.g., circularity, disjoint partition errors, incorrect classifications
- Ontology incompleteness:
  - Imprecisely defined or missing concepts, partially defined disjointness properties, redundancy of class, instance, or relation

Ontologies can be Evaluated per Questions

- Is the ontology mappable to some specific upper ontology, so that its evaluation will be at least partially dependent on the evaluation of the latter also?
- What is the ontology's underlying philosophical theory about reality?
  - ***Idealist:*** reality is dependent on mind or is ultimately mental in nature
  - ***Realist*****:** universals or invariant patterns really exist independently of minds (and observers)
  - **Conceptualist:** universals are neither independently existing nor just names but exist only in human and possibly other animal minds as abstractions from particulars
  - ***Nominalist*****:** only particulars exist and universals do not exist in reality or in our minds but only as general terms
  - ***3-dimensionalist*****:** space and time exist independently and material objects are extended in space and endure through time,
  - ***4-dimensionalist***: only a combined spacetime exists; etc.
- What kinds of reasoning methods can be invoked on the ontology, i.e., by the inference engine that uses it?

Additional Issues for Ontology Evaluation

- Aligning with other existing ontologies, e.g., importing OWL ontologies
  - All the entailments of the imported ontology now hold of the importing ontology
  - Establishing equivalence relations between classes/properties of the importing and imported ontologies
  - Term agreement (assuming the semantics can be read off the term name) is prone to error
- But importing ontologies may introduce inconsistencies, even or especially when equivalences are made between classes/properties of the importing and imported ontologies
- Meta-properties: Transitivity, Reflexivity, Symmetry
  - In OWL these are available axioms, i.e., in addition to Subclass (which is Transitive, Reflexive, Anti-Symmetric), you can define your own properties which have these
  - Are partOf/hasPart properties transitive? Always?
  - Maybe it's better to import an upper/foundational ontology that defines these?
- Defining additional, content-based meta-properties: e.g., OntoClean's determination of "rigidity" value correlation between a parent and a child node in the taxonomic backbone

Task-Based Evaluation of Ontologies: Requirements, etc.

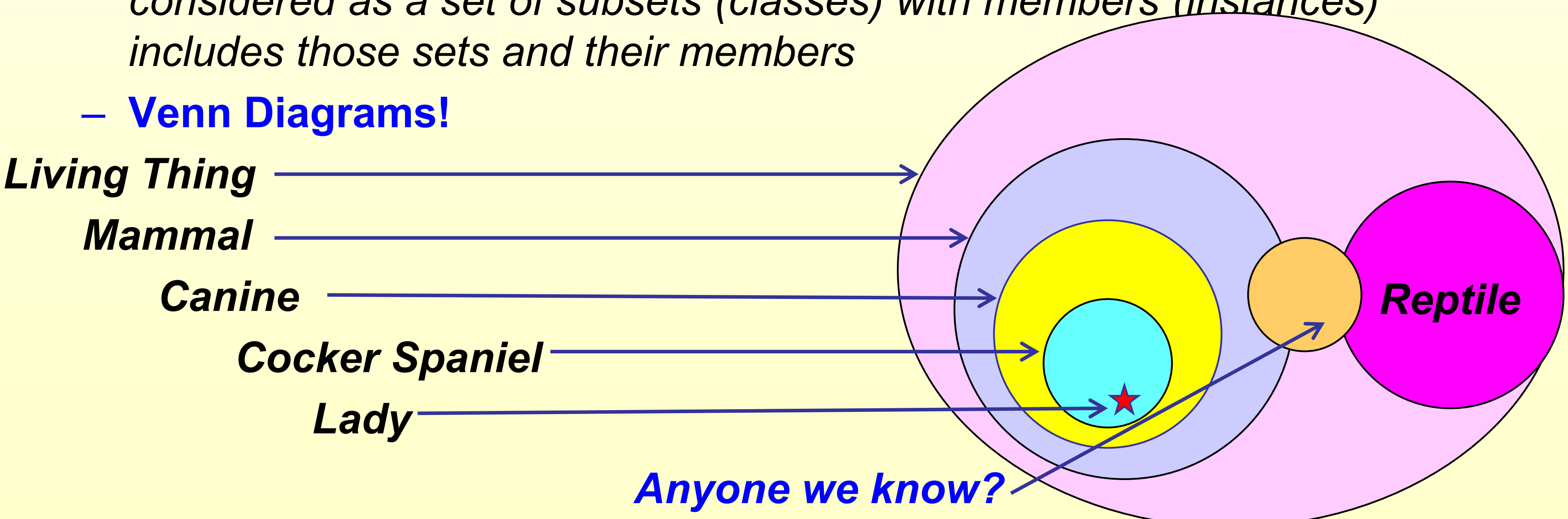
- The human ability to formulate queries using the query language provided by the ontology
- The accuracy of responses provided by the system's inferential component (and there may be more than one)
- The degree of explanation capability offered by the system
- The coverage of the ontology in terms of:
  - The degree of reuse across domains
  - The scalability of the knowledge base
  - The ease of use of the query component
- Are the constructs of the ontology (classes, properties, instances in OWL; predicates and axioms in other KR languages) annotated?
  - Descriptions, comments in natural language about intended meaning, synonyms/antonyms, examples, citations and other provenance information, i.e., alignment suggestion with other ontologies?

Collaborative Ontology Development: Evaluation Issues

- Common practice in large efforts is insulating your ontology module from other simultaneous ontology development
  - May require integrative or “overlay” ontologies that act as integration bridges between the given ontologies
  - Not only importing, but bridging
- Maintenance, redeployment, adding new applications
  - Requires Versioning of all ontologies: not just syntactic, but semantic
  - Periodic retesting of consistency of the ontology modules
  - Regression testing: queries and rules must be tested again and again, to gauge effect, evaluated

Example: Inheritance of Properties, Subsumption

- **Developing a sound taxonomic backbone, i.e., a central subClass subsumption taxonomy is very important: $\forall x\ P(x) \rightarrow Q(x)$**
  - Nearly everything else in the ontology depends on this
  - This is the transitive, reflexive, anti-symmetric classification pipeline
  - Mathematically, it makes the core ontology a partially-ordered set
  - Parent classes subsume children classes
  - Subsumption: usually defined *extensionally, i.e., the parent class when considered as a set of subsets (classes) with members (instances) includes those sets and their members*
  - **Venn Diagrams!**



Example: Inheritance of Properties: Disjoint & Exhaustive Partitioning?

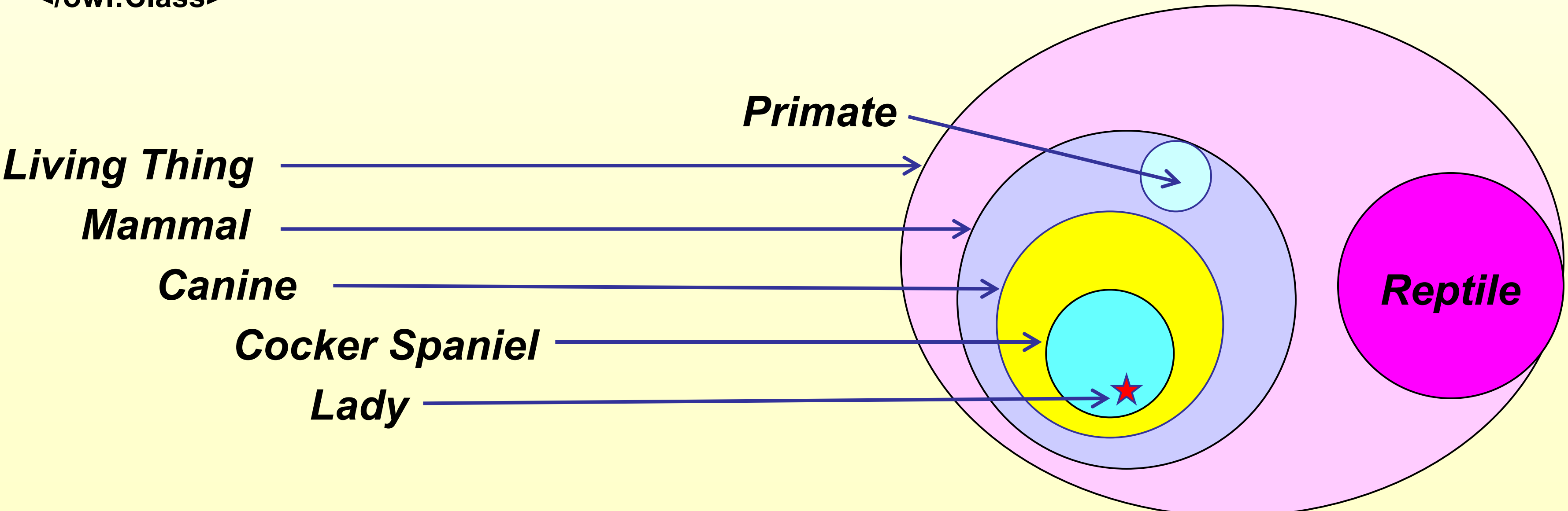
- Disjoint: We can declare by axiom that Canine is Disjoint from Reptile

```
<owl:Class rdf:about="#Canine">
  <rdfs:subClassOf rdf:resource="#Mammal"/>
</owl:Class>

<owl:Class rdf:about="#Primate">
  <rdfs:subClassOf rdf:resource="#Mammal"/>
  <owl:disjointWith rdf:resource="#Canine"/>
</owl:Class>
```

- Exhaustive Partioning: Union AND Disjoint in OWL 2 (*contrived*)

```
<owl:Class rdf:about="#Mammal">
    <rdfs:subClassOf rdf:resource="#LivingThing"/>
    <owl:disjointUnionOf rdf:parseType="Collection">
      <rdf:Description rdf:about="#Canine"/>
      <rdf:Description rdf:about="#Primate"/>
    </owl:disjointUnionOf>
  </owl:Class>
```



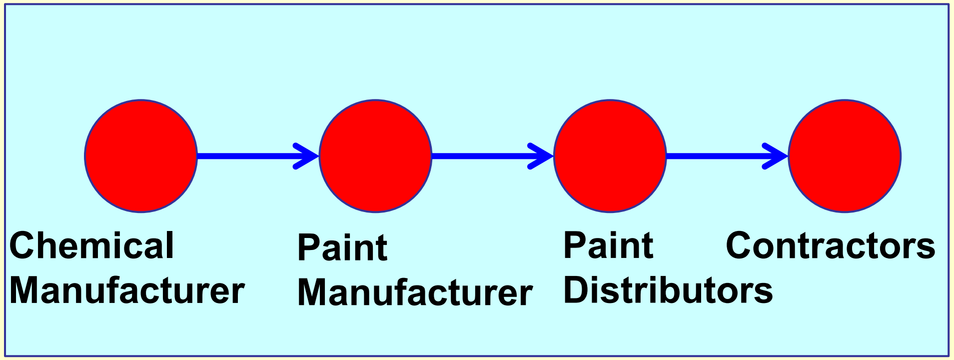
Use of Formal Ontological Analysis: OntoClean & Other Sound Approaches*

- **Ontology Development should be based on the following theories:**
  - **Theory of Parts**: Mereology or mereotopology? Is parthood transitive? Some, some not.
  - **Theory of Wholes**: what is the difference between a part and a whole?
  - **Theory of Essence and Identity**: what are essential, i.e., necessary, properties? If you lose a necessary property, you lose identity. If John loses an arm, he's still John. But if he loses his head? Is he John if he's dead?
  - **Theory of Dependence:** some things and properties depend on others
  - **Theory of Qualities:** features, attributes, qualia, quality spaces?
  - **Theory of Composition and Constitution**: Venus de Milo statue? Gold bar?
  - **Theory of Participation**: a conceptual framework for describing and analyzing communicative phenomena, agency, community, problem-solving, intersects formal pragmatics, speech acts, intents, etc.
  - **Theory of Representation:** how does one thing represent another? Map represent a region? Plan or specification represent real world steps? Artifact represent function?
  - **Theory of Time, Spacetime, and Events:** Events and States, bridging these.

* Adapted from Guarino, N. Multiple tutorials, 2002-2010.

Real Business-to-Business E-Commerce example: Supply Chain Properties (from VerticalNet, 2000)

- Where you are in the supply chain determines the sub-ontology you need
- But you must bridge to your down/upstream supply chain partners
- **Chemical Manufacturer requires:**
  - **Physical classes and properties:**
    - Chemical elements, chemical compounds, chemical reactions, valency, etc.
    - Chemical processes: change or combine chemicals, chemical compounds, but also: chemical manufacturing processes (chemical engineering, etc.)
  - Purity, volatility, etc.
- **Paint Manufacturer requires:**
  - **Functional classes and properties**:
    - Light Reflectivity
    - Drying Time
    - Durability
    - Safety, exposure
    - Shelf life



Conclusion

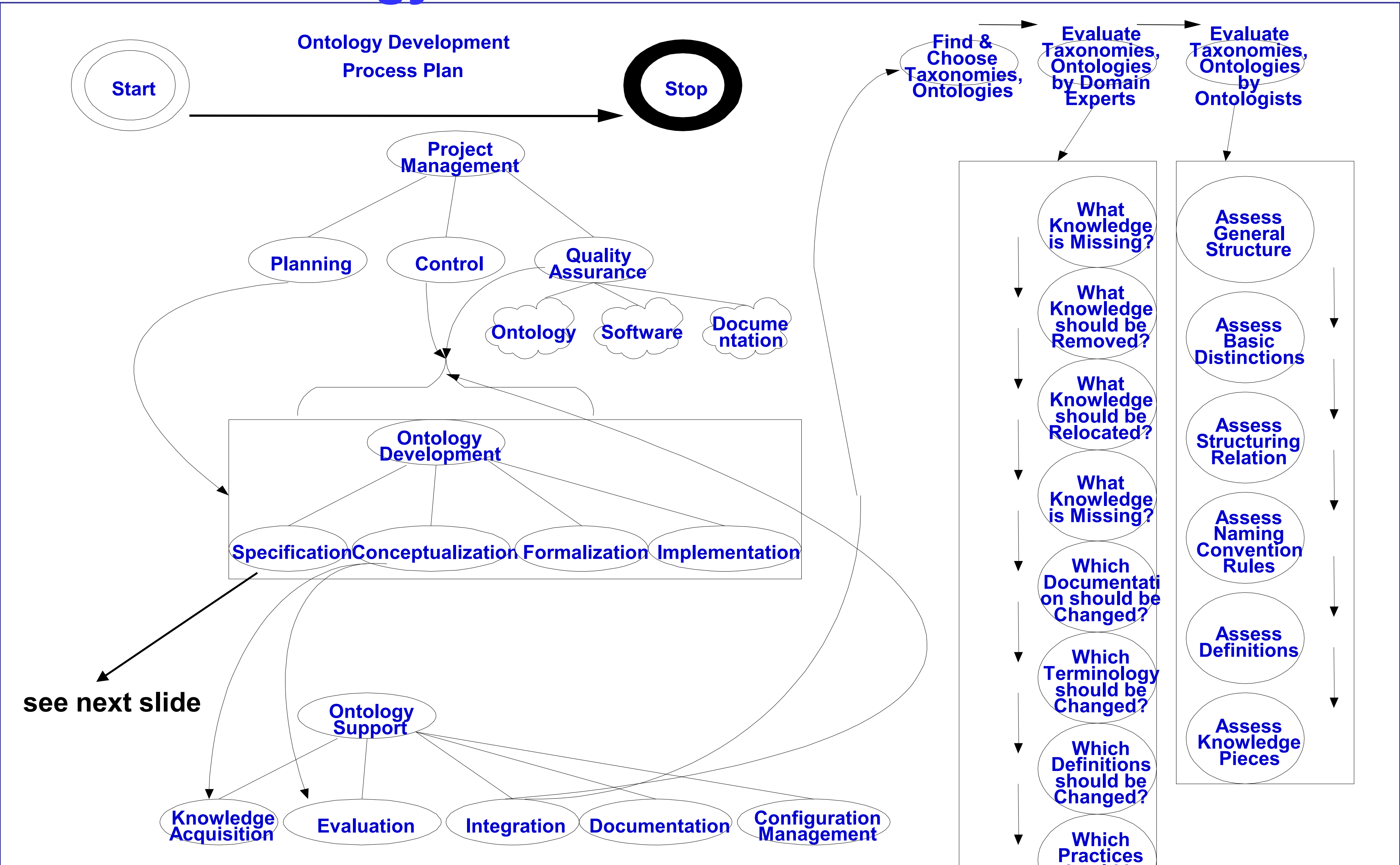
Ontology Evaluation is hard …

… because Quality Ontology Development is hard!

Thanks!

Backup

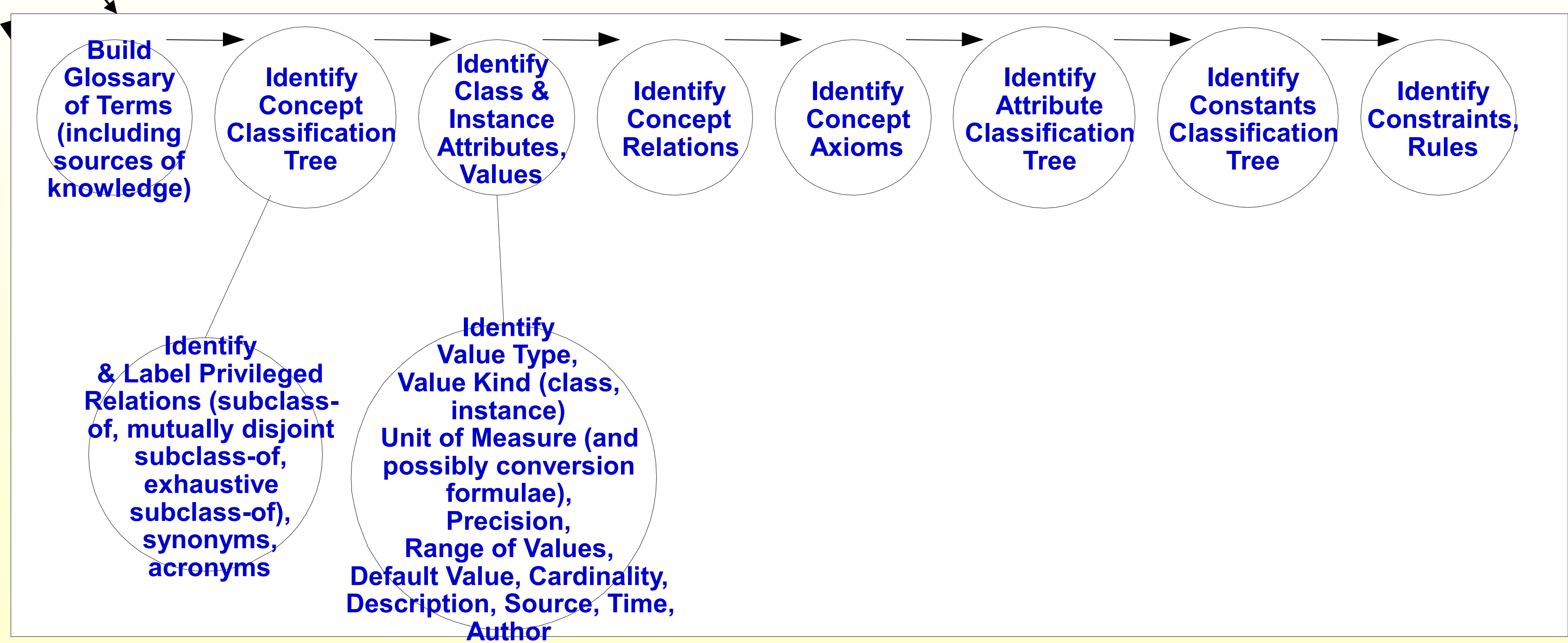
Ontology Development Process Plan: Based on MethOntology



*Fernandéz, Mariano; Gómez-Pérez, Asunción; and Juristo, Natalia. 1997. METHONTOLOGY: From Ontological Art to Ontological Engineering. Workshop on Ontological Engineering. AAAI Spring Symposium Series. Stanford University, pp. 33-40.

Ontology Development Process Plan

from previous slide



Ontology Lifecycle

4) Analysis 3
- What are the resources available to harvest: vocabularies, schemas, taxonomies, conceptual models, ontologies?
- Are there domain standards, upper/middle ontologies to embed what we create within?

3) Analysis 2
- What are the referents, concepts: entities, relations, properties, rules?
- What are the terms that index the referents: terminology?

2) Analysis 1 (Competency Questions)
- *Bottom-Up*: What are semantics of current data sources?
- *Top-Down:* What would you like to ask?

1) Rationale: Why do you need an ontology?

8) Analysis 4
- Refine with domain experts, end users

9) Design 3
- Refine conceptualization

10) Implement 2
- Refine ontology

11) Deploy 1
- Provide ontology application services

12) Deploy 2
- Correct problems

13) Analysis 5
- Interrogate users
- Refine reqs
- More resources?

14) Design 4
- How can changes needed be made?
- Refine reqs

5) Design 1
- What ontology architecture do we choose?
- How expressive is the ontology language we need?
- What conceptualization?
- How do we model these entities, relations, properties, rules?
- What are the instances of these?
- What data sources mappings can link to these? How?
- What kinds of ontology tools do we need?

6) Implement 1
- Implement the ontology server we will need: periodicity, granularity, configuration management
- Implement the infrastructure, services of our architecture: enhance the server with application, SOA support

7) Design 2
- Are we done with ontology development?
- Test competency questions as queries against ontology + data: are good answers returned quickly wrt domain experts/end users?

Ontology Maturity Model

Most Mature

From less to more mature

OMM Level 5

Consistent, pervasive capture of real domain semantics embedded under common middle/upper semantics (axiomatized ontologies); extensive inference

OMM Level 4

Consistent & pervasive capture of real domain semantics, represented as persistent & maintained models (frame ontologies, some axioms); some linkage to upper/middle; some inference supported;

OMM Level 3

Focus is on capture of real domain semantics, mostly represented as persistent & maintained models (frame ontologies); term resources linked to models; database and information extraction routines use some domain ontologies

OMM Level 2

Principled, consistent local semantics captured, some real domain semantics represented as persistent & maintained models (local ontologies); term & concept (referent) distinguished; databases and information extraction routines use local ontologies

OMM Level 1

Mainstream syntactic/structural DB technology (+ data warehouses + data marts), unstructured data addressed by procedural information extraction, no persistent linkage of semantics to syntax/structure, ad hoc local semantics sometimes captured in data dictionary & commented in extraneous code; no clear distinction made between term & concept (referent)

Least Mature