Ontology Summit 2013:
Ontology Evaluation Across the Ontology Lifecycle
Virtual Panel Session 12 – April 4, 2013

Track A: Intrinsic Aspects of Ontology Evaluation: Synthesis 2

Co-Champions:

Steve Ray (Carnegie Mellon University Silicon Valley)
Leo Obrst (MITRE)

Track A: Intrinsic Aspects of Ontology Evaluation: Mission (1)

- Ultimately an ontology's worth can be measured by the effectiveness with which it helps in solving a particular problem
- But as a designed artifact, there are a number of intrinsic characteristics that can be measured for any ontology that give an indication of how "well-designed" it is:
  - Proper use of various relations found within an ontology
  - Proper separation of concepts and facts (sometimes referred to as classes vs. instance distinctions)
  - Proper handling of data type declarations
  - Avoidance of assuming semantics in naming (sometimes called "optimistic naming")
  - Consistent range & domain constraints
  - Better class/subclass determination
  - The sound use of principles of ontological analysis

Track A: Intrinsic Aspects of Ontology Evaluation: Mission (2)

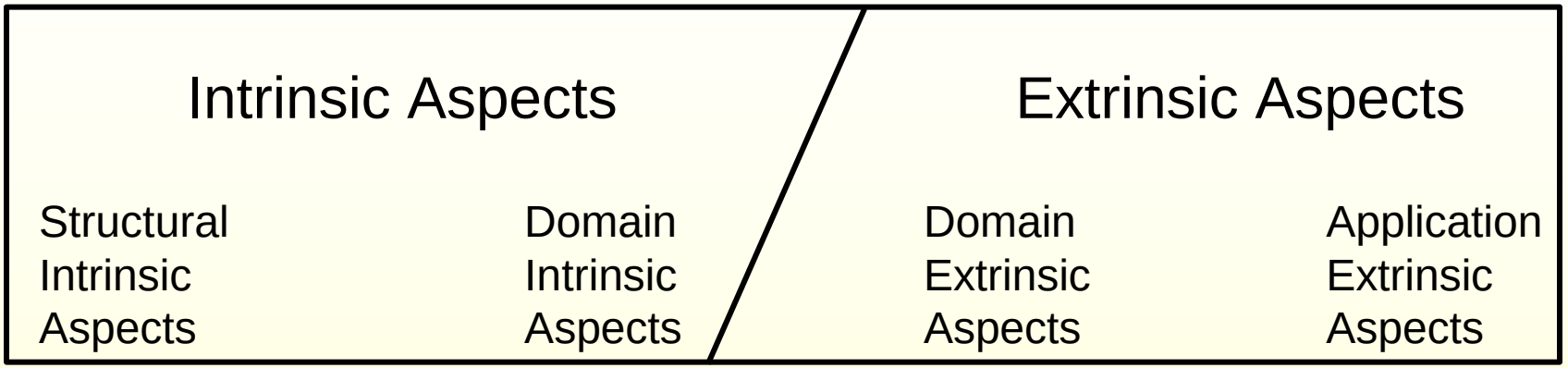
- This Track aims to enumerate, characterize, and disseminate information on approaches, methodologies, and tools designed to identify such intrinsic characteristics, with the aim of raising the quality of ontologies in the future

- Scope:
  - Dimensions of evaluation: structure, logic, semantics, analysis
  - Methods of evaluation
  - Criteria
  - Properties to measure

Intrinsic Aspects

- We focus on the evaluation of ontologies under the following intrinsic aspects
  - Is the ontology free of obvious inconsistencies and errors in modeling?
  - Is the ontology structurally sound? How do we gauge that?
  - Is the ontology appropriately modular?
  - Is the ontology designed and implemented according to sound principles of logical, semantic, and ontological analysis?
  - Which intrinsic aspects of ontology evaluation are of greater value to downstream extrinsic ontology evaluation?
  - Intrinsic Aspects Comparable to White/Glass Box Testing?

Partitioning the Ontology Evaluation Space: Intrinsic to Extrinsic



- Intrinsic ontology evaluation, from our perspective, consists of two parts:
  - *Structural Intrinsic Evaluation*
  - *Domain Intrinsic Evaluation*

Structural Intrinsic Evaluation

- Ontology evaluation that does not depend at all on knowledge of the domain being modeled
- Does draw upon mathematical and logical properties such as graph-theoretic connectivity, logical consistency, model-theoretic interpretation issues, inter-modularity mappings and preservations, etc.
- Structural metrics such as branching factor, density, counts of ontology constructs, averages, and the like are intrinsic
- Some meta-properties such as transitivity, symmetry, reflexivity, and equivalence may also figure in intrinsic notions
- In general, structural intrinsic criteria are focused only on domain-independent notions, mostly structural, and those based on the knowledge representation language

Domain Intrinsic Evaluation

- Evaluation where some understanding of the domain is needed in order to, for example, determine that a particular modeling construct is in alignment with the reality it is supposed to model
- It may be that some meta-properties such as rigidity, identity, unity, etc., suggested by metaphysics, philosophical ontology, semantics, and philosophy of language are used to gauge the quality of the axioms of the ontology, including e.g., the subclass/isa taxonomic backbone of the ontology and other structural aspects of the ontology
- Most of the aspects of this category focus on ontological content methods such as better ontological and semantic analysis, including meta-property analysis (such as provided by methodologies like OntoClean, etc.)
- Domain knowledge and better ways to represent that knowledge do come into play here, though divorced as much as possible from application-specific domain requirements that come more explicitly from extrinsic evaluation issues.
- At the extrinsic edge of domain intrinsic evaluation, the context-independent measures from Structural Intrinsic evaluation begin to blend into the very context-dependent, application issues of Extrinsic evaluation

Structural Intrinsic Evaluation Tools

- OOPS!: Reports on suspected improper uses of various OWL DL modeling practices
  - http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp
  - Described by MariaPovedaVillalon
- OntoQA to develop metrics for any ontology based on structural properties and instance populations
  - Described by SamirTartir

Domain Intrinsic Evaluation Tools

- The OQuaRE framework combines both context dependent and independent metrics
  - Described by AstridDuqueRamos
  - The OQuaRE team has stated their desire to better distinguish between these two categories of metrics

- The OntoClean methodology
  - Not reported on in Ontology Summit 2013, but generally well-known [1, 2]
  - Draws upon meta-domain knowledge, the use of meta-properties, i.e., standard evaluative criteria originating from the practices of ontological analysis

[1] N. Guarino, C. Welty. 2002. Evaluating Ontological Decisions with OntoClean. Communications of the ACM. 45(2):61-65. New York: ACM Press. http://portal.acm.org/citation.cfm?doid=503124.503150.
[2] Guarino, Nicola and Chris Welty. 2004. An Overview of OntoClean. In Steffen Staab and Rudi Studer, eds., The Handbook on Ontologies. Pp. 151-172. Berlin:Springer-Verlag. http://www.loa-cnr.it/Papers/GuarinoWeltyOntoCleanv3.pdf.

Evaluation Across the Ontology Lifecycle

- Every criterion should be evaluated at each point in the ontology lifecycle, but with some criteria being more important (necessary/sufficient) at some points more than others
- Therefore, a better ontology evaluation methodology might define *necessary and sufficient criteria* (and their measures) derived from both intrinsic and extrinsic aspects that apply to different points in the ontology lifecycle
- In addition, the determination of these necessary or sufficient criteria may be subject to constraints:
  - For example, though initially an intrinsic criterion of logical consistency of the ontology may be imposed as a necessary property at the beginning of the first phase of ontology development, it might be relaxed subsequently when it is determined that a different semantics will apply in how the ontology is interpreted within a given application
  - E.g., if the application-specific reasoning will not observe the full FOL or description logic Open World Assumption, but instead interpret the ontology under a Closed World Assumption)

Ontology Lifecycle

1) Rationale: **Why do you need an ontology?**

2) Analysis 1 (Competency Questions)

- ***Bottom-Up*: What are semantics of current data sources?**
- ***Top-Down:* What would you like to ask?**

3) Analysis 2

- **What are the referents, concepts: entities, relations, properties, rules?**
- **What are the terms that index the referents: terminology?**

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?**

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?**

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**