



**Ontology Summit 2013**  
**Track B**  
**Recap**  
**and**  
**Questions for Consideration**

**Terry Longstreth**  
**Todd Schneider**

**2 May 2013**

# Mission Statement



The intent is to explore, clarify, and identify gaps, practical and theoretical, in the of evaluation of ontology from a systems perspective using the paradigm of blackbox evaluation.

- Extrinsic aspects of ontology evaluation includes subjective factors, measures or metrics, and the range of values of quantifiable attributes.
- In a systems context evaluations are derived from examination of inputs or stimuli (to the blackbox) and the outputs or externally measurable attributes or behaviors, where those behaviors are controlled or influenced by an ontology.
- The ontology in question may be fully embedded/encapsulated within an entity or system, or may be externally accessible (and potentially shared) among multiple entities or systems.
- The separation of system or entity behaviors which are not governed by an ontology must be accounted for in any ontology evaluation process.

# Synthesis Recap - 1



## Scope

- Ontology evaluation takes place in some context - That context imparts a scope (of applicability or validity).
- Evaluation context and scope directly impacts the evaluation criteria
- Evaluation criteria values/ranges differ depending on evaluation context, scope, lifecycle phase(s)
- Most important evaluation context - how an ontology is expected to be used and in what operational domains

## Reuse

- Many of the systems and software paradigms and processes can be used for the evaluation of ontologies.
- Relational databases bear strong similarity to ontologies
- Techniques of database evaluation and testing can be applied to ontology evaluation.

# Synthesis Recap - 2



## Security

- Security vulnerabilities of systems using ontologies may exist
- May impact validity of embedded ontology.
- Ontologies themselves may pose no security risks, but infrastructure (e.g., reasoners, triple stores, etc.), system interfaces may introduce security vulnerabilities. (e.g., RDB example - SQL injection attack)
- Security and evaluation methods or techniques NOT adequately addressed
- An open area for research.

## Automation

- Software engineering has tools that automate parts evaluation over the lifecycle (e.g., regression analysis)
- Ontology evaluation has few such tools for any part of lifecycle
- Projects underway that may automate evaluation of some criteria.

# Synthesis Recap - 3



## Dynamics

- Apparent implicit assumption that ontology / theory are static (during operations phase)
- Ontologies can provide a system with dynamic capabilities
- During system operations ontology can change
- Existing systems and software evaluation paradigms may not be applicable.
- Ontology evaluation(s) don't explicitly address dynamics of ontology, hence evaluations thereof.
- How does ontology evaluation change for dynamic ontologies?
- Simulation(s) may provide paradigm to fill this gap (e.g., NEMO project)

# Synthesis Recap - 4



## Requirements – Basis for Evaluation

- Requirements are extrinsic to ontology and are not design
- Lack consistent capability to create specifications that
  - Represent semantic requirements and operational needs
  - Describe intended interpretations/models
  - Can be easily translated into tests
- Without adequately expressive requirements for ontology, any evaluation will fail to meet expectations
- Ontology requirements need to be derivable from general systems requirements
- Need requirements development process that meet needs of
  - Ontology evaluation across the lifecycle and have appropriate lifecycle level of detail
  - Potential complexity of systems employing ontologies
  - Testing and existing testing paradigms
  - Automated testing and evaluation
- Establishing reliable methods and tools for ontology requirements development will
  - Provide ability to create testable criteria
  - Improve ontology development and use,
  - Couple ontology use to deliverable system(s).

# Questions



1. Does evaluation of an ontology vary with the purpose of the ontology?
  - If so, how?
2. How can requirements for ontology be crafted/derived from general systems or business requirements?
  - Are there general principles, patterns, or practices for deriving ontology requirements from system requirements?
3. How do requirements fit into a [ontology] development methodology?
  - Is Ontology requirements management different as a discipline from its counterpart in systems engineering?