Ontology Summit 2013 Track B Extrinsic Evaluation

Synthesis & Communiqué Themes

Terry Longstreth
Todd Schneider

21 February 2013

24 Jan – First Panel Established Ground Rules

- Todd Schneider Overview and Introduction
  - Use of blackbox evaluation techniques to ontology evaluation
  - Establish boundaries of utility for blackbox evaluation
  - Eventual goal: organize evaluation dimensions
- Hans Polzer
  - Context dimensionality reflects impact of evaluation context on evaluation processes and tools
  - Evaluation Scope, Context, Purpose, Lifecycle Phase, Application domain
  - Ontology evaluation attributes for applicable context dimensions and ranges

24 Jan – Continued

- Mary Balboni
  - Software Development as a model for Ontology preparation
  - Blackbox evaluation used throughout SW development lifecycle
    - Each test case includes expected results (behaviors)
    - Test data (driving test cases) used to probe limits of test domain
  - Blackbox testing may be improved with fault seeding
- Megan Katsumi
  - Methodology for Development and Verification of Expressive Ontologies
  - Evaluation and Verification of full first order ontologies
  - Posits evaluation scope tied directly to formal specification
  - Might be useful for evaluating equivalence of common concepts between paired or partially paired ontologies
    - Ontology used for developer guidance and embedded operational ontology
    - Regression testing across versions

Track B Synthesis

Evaluation Scope Driven By Intended Use

- The purpose served by an ontology is the paramount determinant of criteria and their metrics pertinent to evaluation of that ontology

  As corollary,

  Evaluation of an ontology w.r.t. a lifecycle phase cannot be the sole determinant of the ontology's utility for any other lifecycle phase

  - There is a utility function associated with an ontology in each lifecycle phase which governs the ontology evaluation scope for that phase
  - The Utility Function is different for development/engineering aids (ex ISO 15926), operational guidance (cf. ITIL), machine inferencing
  - Ontology lifecycle phases may alter utility function for an ontology

  At a minimum – several qualitatively different lifecycle phases, from initial need recognition through development, deployment, management and maintenance, to retirement and disposition (whether archiving or destruction)

- Enterprise Lifecycle phases are additional cross-cutting domains
- Architectural models of ontology and implementations greatly impact extrinsic evaluation issues.

Other Contributing Sources

Mike Bennett -**Track C**

Made clear that there were sometimes more than one ontology at issue.

- Conceptual Model (his term) – independent of implementation technology
- Operational Ontology – implementation tied back to conceptual model
- "The formal methodology needs to define what tools and techniques are applied at what points in the written development process, to what artifacts, to what end."

- Other Track C and A sources need deeper review, particularly for Architectural perspective in Synthesis

Scope and context keywords 24 Jan – Track B – Chat Gleanings

Property / Characteristic	Full Text
Reusefulness,	**JackRing:** Reusefulness of an ontology or subset(s) thereof?
Quality, Parsimony, Beauty	**JackRing:** In systems think the three basic dimensions are Quality, Parsimony, Beauty
License	**MariCarmenSuarezFigueroa:** In the legal part, maybe we should consider also license (and not only copyright)
Relevance, Clarity, Consistency, Accessibility, timeliness, completeness, accuracy, costs (development, maintenance), Benefits, Provenance, Modularity	**MatthewWest:** Relevance, Clarity, Consistency, Accessibility, timeliness,completeness, accuracy, costs (development, maintenance), Benefits **MatthewWest:** Provenance **MatthewWest:** Modularity
complexness	**JackRing:** No one has mentioned the dimension of complexness. Because ontologies quickly become complex topologies then the response time becomes very important if implemented on a von Neumann architecture. Therefore the structure of the ontology for efficiency of response becomes an important dimension
Reliability, Availability, Maintainability, Performance, Scalability, Security.	**BobbinTeegarden:** At DEC, we used an overlay on all engineering for RAMPSS -- Reliability, Availability, Maintainability, Performance, Scalability, and Security. Maybe these all apply for black box here? Mary has cited some of them...
domain integrity, referential integrity, semantic integrity, Precision With_Respect_To(domain D, requirement R)	**LeoObrst6:** @MaryBalboni: re: slide 14: back in the day, we would characterize 3 kinds of integrity: 1) domain integrity (think value domains in a column, i.e., char, int, etc.), 2) referential integrity (key relationships: primary/foreign), 3) semantic integrity (now called "business rules"). Ontologies do have these issues. On the ontology side, they can be handled slightly differently: e.g., referential integrity (really mostly structural integrity) will be handled differently based on Open World Assumption (e.g., in OWL) or Closed World Assumption (e.g., in Prolog), with the latter being enforced in general by integrity constraints. **LeoObrst6:** @Todd: your second set of slides, re: slide 4: Precision, Recall, Coverage, Correctness and perhaps others will also be important for Track A Intrinsic Aspects of Ontology Evaluation. Perhaps your metrics will be: Precision With_Respect_To(domain D, requirement R), etc.? Just a thought. **LeoObrst6:** Perhaps the main difference between Intrinsic -> Extrinsic is that at least some of the Intrinsic predicates are also Extrinsic predicates with additional arguments, e.g., Domain, Requirement, etc.?

Chat Gleanings continued

Effectiveness, Beauty	**BobbinTeegarden:** @JackRing Would 'effectiveness' fall under beauty? What criteria? **JackRing1:** @Bobbin, Effect-iveness is a Quality factor. Beauty is in the eye of the beer-holder
Requirements Satisfaction	**MariCarmenSuarezFigueroa:** We could also consider the verification of requirements (competency questions) using e.g. SPARQL queries.
consistency; correctness, completeness	**TillMossakowski:** further dimensions: consistency; correctness w.r.t. intended models (as in Megan's talk), completeness in the sense of having intended logical consequences
Goodness Elegance	**BobbinTeegarden:** It seems we have covered correctness, precision, meeting requirements, etc well, but have we really addressed 'goodness' of an ontology? And certainly haven't addressed an 'elegant' ontology, or do we care? Is this akin to Jack's 'beauty' assessment?
Simplicity Minimality Normalized	**BobSchloss:** Because of the analogy we heard with Database Security Blackbox Assessment, I wonder if there is an analogy to "normalization" (nth normal form) for database schemas. Is some evaluation criteria related to factoring, simplicity, minimalism, straightforwardness.....
Granularity Update Impedance/ complexity/ cost Degree of stability Error Discovery Profile	**TorstenHahmann:** another requirement that I think hasn't been mentioned yet: granularity (level of detail) **BobSchloss:** I am also thinking about issues of granularity and regularity ... If a program wants to remove one instance "entity" from a knowledge base, does this ontology make it very simple to just do the remove/delete, or is it so interconnected that removal requires a much more complicated syntax.... **LeoObrst6:** @Torsten: yes, that was my question, i.e., granularity. **MariCarmenSuarezFigueroa:** I'm also think granularity is a very important dimension.... **BobSchloss:** Although this is driven by the domain, some indication of an ontology's rate of evolution or degree of stability or expected rate of change may be important to those using organizations. If there are 2 ontologies, and one, by being very simple and universal, doesn't have as many specifics but will be stable for decades; whereas another, because it is very detailed using concepts that are related to current technologies, current business practices, and therefore may need to be updated every year or two... I'd like to know this. **BobSchloss:** Another analogy to the world of blackbox testing... the software engineers have ideas of Orthogoal Defect Classification and more generally, ways of estimating how many remaining bugs there are in some software based on the rates and kinds of discovery of new bugs that have happened over time up until the present moment. I wonder if there is something for an ontology... one that has a constant level of utilization, but which is having a decrease in reporting of errors.... can we guess how many other errors remain in the ontology? Again... this is an analogy.... some way of estimating "quality"... **MatthewWest:** Yes, stability is an important criteria. For me that is about how much the existing ontology needs to change when you need to make an addition.