This publication is available free of charge from http://dx.doi.org/10.6028/NIST.IR.8008

NISTIR 8008

Toward Ontology Evaluation across the Life Cycle

edited by
Fabian Neuhaus
Steve Ray
Ram D. Sriram

http://dx.doi.org/10.6028/NIST.IR.8008

National Institute of Standards and Technology
U.S. Department of Commerce

This publication is available free of charge from http://dx.doi.org/10.6028/NIST.IR.8008

NISTIR 8008

Toward Ontology Evaluation across the Life Cycle

edited by

Fabian Neuhaus
Otto-von-Guericke University
Magdeburg, Germany

Steve Ray
Carnegie Mellon University
Moffett Field, CA

Ram D. Sriram
Software and Systems Division
Information Technology Laboratory

This publication is available free of charge from
`http://dx.doi.org/10.6028/NIST.IR.8008`

October 2014

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Acting Under Secretary of Commerce for Standards and Technology and Acting Director

This publication is available free of charge from http://dx.doi.org/10.6028/NIST.IR.8008

Disclaimer

No approval or endorsement by NIST of any commercial product or service is intended or implied. Certain commercial companies, software tools, and services are identified in this report to facilitate better understanding. Such identification does not imply recommendation or endorsement by NIST nor does it imply those identified are necessarily the best available for the purpose.

Table of Contents

Section 2: Toward Ontology Evaluation across the Life Cycle

Section 3: Going Forward

Introduction

Fabian Neuhaus, Steve Ray, and Ram D. Sriram

Ontology is an information technology that enables computers to 'understand' and reason with complex knowledge. This technology has been shown to have a wide range of applications; including data mining, natural language processing, information federation and integration, and the use within standards to represent concepts unambiguously.

In spite of its success, there is a lack of methods, metrics, and tools for improving the process of ensuring that ontologies are designed well and that they behave correctly within the context of the information systems that they are part of. This lack of measurement science is an obstacle for industry to improve the quality of ontology development and maintenance.

The goal of the Ontology Summit 2013 was to identify existing best practices and tools in ontology evaluation, and to create a community consensus on how to best ensure quality across the whole lifecycle of an ontology. This report is designed to both report on the 2013 Ontology Summit, and to interpret the results of the summit in order to better guide the National Institute of Standards and Technology in its future strategic positioning with respect to semantic technology.

The first part of this report summarizes each section of the 2013 summit, composed of the four thematic tracks and a series of hackathons. The second part contains the Summit Communiqué that represents the group's position on the state of ontology evaluation. The third part contains some observations and recommendations for NIST to consider when deciding on the path forward in supporting the semantic web, and in its broader mission of supporting high quality standards.

Section 1

Requirements and Methods for Ontology Evaluation

Background and Organization of the Ontology Summit 2013

Steve Ray

1 The Ontology Summit Series

Increasingly, major national and international projects centered on ontology technology are being advanced by governments and by scientific and industrial organizations. In 2002, ONTOLOG (a.k.a. 'Ontolog Forum') was created as 'an open, international, virtual community of practice devoted to advancing the field of ontology, ontological engineering and semantic technology, and advocating their adoption into mainstream applications and international standards.'

In 2006, that community teamed up with National Institute of Standards and Technology (NIST) and initiated an annual, goal-oriented gathering of the top ontologists in the world. This gathering was called the Ontology Summit. Since that time, the Ontology Summit series has teamed up with more co-organizers, broken new ground in both content and process, and has successfully brought the academic, industrial, and standards communities together. The Summit series has also grown in participation each year. A compilation of participation statistics can be found for 2011, 2012 and 2013 at [1], [2], and [3] respectively, showing double-digit increases in subscribers to the discussion list each year. Ontological engineering and ontological analysis, with its technical rigor and expressiveness, is showing great promise and some concrete results in each of these sectors, particularly in the areas of semantic data integration and information standards. Themes for all the Ontology Summits are:

2006 – Upper Ontologies [4]
2007 – Ontology, Taxonomy, Folksonomy: Understanding the Distinctions [5]
2008 – Toward an Open Ontology Repository [6]
2009 – Toward Ontology-based Standards [7]
2010 – Creating the Ontologists of the Future [8]
2011 – Making the Case for Ontology [9]
2012 – Ontology for Big Systems [10], [11]
2013 – Ontology Evaluation across the Ontology Lifecycle [12]

These themes were chosen because they touched on topics that affect ontology research and development in all application domains. They speak to common issues, such as the need to bridge between ontologies and whether upper ontologies can play a part; the confusion that exists in the public over just what an ontology is; the need to find and manage the increasing number of mature ontologies; the potential for the use of ontology technology for normative information standards; the need to train the next generation of ontologists; the application of ontologies in the development and exploitation of 'big systems' and 'big data;' and the tools and methods for ensuring quality in ontologies and ontology-based systems.

2 2013 Ontology Summit – Ontology Evaluation across the Ontology Lifecycle

The 2013 Ontology Summit was titled 'Ontology Evaluation across the Ontology Lifecycle' and sought to explore, identify and articulate the tools and methods in use, and often freely available, for ensuring the quality of ontologies under development or in use. As described in the communiqué below, the evaluation process was discussed for each of eight phases of an ontology life cycle, from requirements development, through design, to ultimate operation and maintenance. Further, the evaluation techniques ranged from structural metrics such as the branching factor of an ontology (part of what was called 'Craftmanship'), to evaluation of fitness to a problem, performance of a system, and utility in solving a problem. As is traditional with the Ontology Summit series, the conclusions of the presentations and discussion were captured in the form of a communiqué, with expanded supporting material provided on the web. Furthermore, the traditional wiki site [13] is augmented with the more presentable audience-facing site [12] – a practice that began in 2012.

2.1 Chronology

The ontology summit series has refined a highly effective program design which draws upon a number of collaboration techniques in order to produce a concrete result on a fixed schedule, even though the participants are a widely distributed, self-selecting group of individuals. The summit consists of:

- A roughly three-month virtual discourse on the summit theme via email discussion lists [14]
- Weekly synchronous virtual meetings consisting of presentations, panels and discussions, supported by teleconferencing, shared screen, and chatroom facilities [15], [16]
- A research study and survey [17]
- A series of hackathons and clinics [18]

- A physical 2-day meeting (held at NIST in Gaithersburg, MD) [19] for final consensus and endorsement of the summit conclusions, documented as a public communiqué [20]. It should be noted that while the communiqué is a good, concrete end-result, the full value of the summit resides in the communiqué plus all the supporting synthesized documentation, email discussion threads, and recordings.
- A collaborative effort in building an ongoing 'community library' pertinent to the summit theme [21].

2.2 The Executive Committee of the Ontology Summit 2013

Each summit theme has contained sub-themes, and to address this, the Ontology Summit Community have used the idea of topic champions to ensure each aspect is properly addressed. For the 2013 summit, we created and filled the following roles:

Chairs	
General Chairs	Matthew West and Michael Gruninger
Symposium Chairs	Mike Dean and Ram D. Sriram
Track Champions	
Track A: Intrinsic Aspects	Leo Obrst and Steve Ray
Track B: Extrinsic Aspects	Terry Longstreth and Todd Schneider
Track C: Building Ontologies	Matthew West and Mike Bennett
Track D: Software Environments	Michael Denny, Ken Baclawski, Peter Yim
Other roles	
Hackathon and Clinics Activities	Mike Dean, Ken Baclawski and Peter Yim
Community Library	Amanda Vizedom
Website	Marcela Vegetti and Ali Hashemi
Communiqué Lead Editors	Amanda Vizedom and Fabian Neuhaus

Co-organizing Institutions

Ontolog
National Institute of Standards and Technology (NIST)
National Center for Ontological Research (NCOR)
National Center for Biomedical Ontology (NCBO)
International Association for Ontology and its Applications (IAOA)
The National Coordination Office (NCO)

2.3 Supporting Committees

Of course, a three month meeting of many world experts does not simply come together naturally. To carry off such an undertaking took months of planning on the part of the principals, and tremendous support by an Organizing Committee and an Advisory Committee. The Organizing Committee was a group of individuals who took on various responsibilities, from championing the technical tracks, editing documents and synthesizing the discussions and conclusions. The Advisory Committee provided specialized expertise as needed during the course of the Summit.

2.4 Summit Proceedings

The next ten sections of this report describe, in sequence, the four thematic tracks that were identified as distinct aspects of ontology evaluation, and the six so-called hackathons that emerged as viable subjects for intensive discussion and development during the three-month summit.

Links

[1] http://ontolog.cim3.net/file/work/OntologySummit2011/2011-05-19_OntologySummit2011_follow-up/OntologySummit2011_statistics--PeterYim_20110518.pdf
[2] http://ontolog.cim3.net/file/work/OntologySummit2012/2012-04-26_OntologySummit2012_follow-up/OntologySummit2012_statistics--PeterYim_20120426.pdf
[3] http://ontolog.cim3.net/file/work/OntologySummit2013/2013-05-23_OntologySummit2013_follow-up/OntologySummit2013_statistics--PeterYim_20130523.pdf
[4] http://ontolog.cim3.net/cgi-bin/wiki.pl?UpperOntologySummit
[5] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2007
[6] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2008
[7] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2009
[8] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2010
[9] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2011
[10] http://ontolog.cim3.net/OntologySummit/2012/
[11] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2012
[12] http://ontolog.cim3.net/OntologySummit/2013/
[13] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013
[14] http://ontolog.cim3.net/forum/ontology-summit/2012-12/index.html
[15] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013
[16] http://ontolog.cim3.net/OntologySummit/2013/OntologySummit2013_ConsolidatedPresentationQuickReference_20130523.html
[17] http://ontolog-02.cim3.net/wiki/Category:OntologySummit2013_Survey

[18] http://ontolog-02.cim3.net/wiki/OntologySummit2013_Hackathon_Clinics
[19] http://ontolog-02.cim3.net/wiki/OntologySummit2013_Symposium
[20] http://ontolog.cim3.net/OntologySummit/2013/communique.html
[21] http://www.zotero.org/groups/ontologysummit2013/items

Track A: Intrinsic Aspects of Ontology Evaluation

Leo Obrst and Steve Ray *

1 Mission Statement

Ontologies are built to solve problems, and ultimately an ontology's worth can be measured by the effectiveness with which it helps in solving a particular problem. Nevertheless, 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. Examples include the proper use of various relations found within an ontology, proper separation of concepts and facts (sometimes referred to as class vs. instance distinctions), proper handling of data type declarations, embedding of semantics in naming (sometimes called 'optimistic naming'), inconsistent range or domain constraints, better class/subclass determination, the use of principles of ontological analysis, and many more. 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.

2 Scope

This document has as scope the dimensions of ontology evaluation, methods, criteria, and the properties to measure to ensure better quality ontologies. We focus in the communiqué 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?

* This article does not contain technical data as defined by the International Traffic in Arms Regulations, 22 CFR 120.10(a), and is therefore authorized for publication.

3 Partitioning the Ontology Evaluation Space

3.1 Intrinsic Evaluation Aspects

Intrinsic ontology evaluation, from our perspective, consists of two parts: Structural Intrinsic Evaluation and Domain Intrinsic Evaluation (see Fig. 1).

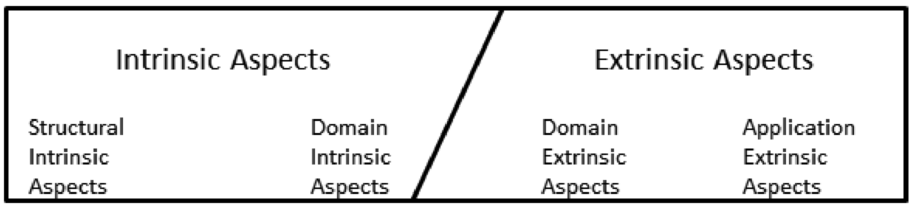

Fig. 1. Structural Intrinsic and Domain Intrinsic Aspects

Structural Intrinsic Evaluation. Ontology evaluation that does not depend at all on knowledge of the domain being modeled, but 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 adherence to implications of transitivity, symmetry, reflexivity, and equivalence assertions 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. Some examples of tools and methodologies that address intrinsic ontology evaluation:

- OOPS! Evaluation described by Maria Poveda Villalon [1]
- OntoQA to develop metrics for any ontology based on structural properties and instance populations, described by Samir Tartir
- Patrick Lambrix's debugging of Isa-a taxonomic structures, especially with mappings between ontologies
- Macleod for automatically checking the consistency, detecting invalid vocabulary terms, and determining provability of competency questions in Common Logic ontologies, as used in Thorsten Hahmann's PhD dissertation.

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. Some examples of tools and methodologies that address domain intrinsic ontology evaluation:

1. OQuaRE framework described by Astrid Duque Ramos
2. OntoClean (Guarino and Welty)
3. Maria Copeland: Ontology Evolution and Regression Testing
4. Melissa Haendel: Ontology Utility from a biological viewpoint
5. Ed Barkmeyer: Recommended practices with mapping vocabularies (especially code-lists) to ontologies.

3.2 Extrinsic Evaluation Aspects

Extrinsic ontology evaluation focuses on the case where the structure and design of the ontology is opaque to the tester, and the evaluation is determined by the correctness of answers to various interrogations of the model. In general, application requirements and domain requirements that are specifically needed by particular applications are the focus of extrinsic evaluation.

4 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. In other words, 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 description logic Open World Assumption, but instead interpret the ontology under a locally Closed World Assumption).

5 More Material

Much of this section was created based upon the excellent presentations made during the summit, which can be found at [2] and [3].

Links

[1] http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp
[2] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_01_31
[3] http://ontolog.cim3?.net/cgi-bin/wiki.pl?ConferenceCall_2013_03_07

Track B: Extrinsic Aspects of Ontology Evaluation

Terry Longstreth and Todd Schneider

1 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*. Extrinsic aspects to be considered include,

- interoperability among ontologies
- requirements and their verification
- how metrics can be derived from requirements
- how 'good' requirements relevant to ontology can be crafted
- fitness for purpose
- query performance
- relevant relational database evaluation methods, metrics and techniques
- differences in evaluation among an ontology and instance data
- how evaluation metrics can be derived from examination of test inputs or stimuli
- how evaluations can be used to revise requirements
- how evaluations can be used to correct an ontology

2 Synthesis

Track B of the 2013 Ontology Summit, Extrinsic Aspects of Ontology Evaluation, focused on aspects of ontology evaluation that did not require direct interaction with, or knowledge of, the ontology. A black-box approach to such evaluation was adopted. The blackbox paradigm properly represents the intent of testing or evaluating extrinsic attributes or qualities of an ontology.

Extrinsic ontology evaluation is a process or group of processes which evaluate ontological commitments in a blackbox mode against specifications [1]. Extrinsic aspects of ontology evaluation include subjective factors, measures or metrics, and the range of values of quantifiable attributes that become available for evaluation over the course of the engineering and operations lifecycles. 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 [2], An Ontology may be fully embedded or 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 ontology must be accounted for in any ontology evaluation process. Much of extrinsic evaluation for ontology is indistinguishable from higher levels of system or enterprise testing models. When properly defined, the evaluating functions are not sensitive to specific ontological commitments or syntax.

As was noted during the summit, the boundary of the blackbox varies across lifecycle phases and evaluation processes [3], [4]; the boundary of the blackbox (for evaluation) expands as development moves to completion. This view allows many of the systems and software engineering processes, paradigms, and techniques which have been developed over the last several decades for evaluation across the engineering lifecycle, to be applied with little or no modification (e.g., unit, sub-system, integration, and [full] system testing). These systems and software paradigms are usually based on requirements or specifications (of different levels of detail).

3 Reuse

During the course of the presentations for Track B we learned that many contemporary systems and software paradigms and processes can be used for the evaluation of ontologies. Database implementations bear a strong similarity to ontologies and their related knowledge bases (i.e., instance data) and have an overlap in the fragment of first order logic applicable. So it's no surprise that techniques of database evaluation and testing, validating the contents, schema, and functionality within a database, can be applied to ontology evaluation. Ultimately we discovered that beyond database or datasystems parallels, the full breadth and depth of 'classical' systems engineering context supplied useful evaluation paradigms, methods and analyses.

Consequently, extrinsic factors in need of evaluation are error conditions, degraded mode operations, load and capacity testing/performance (critical for systems that provide services to a larger enterprise). Validation of the ontology and instance data integrity using system interfaces (e.g., create, read, update, and delete operations) should be performed.

4 Security

Of particular note are issues involving security vulnerabilities (of the system using ontologies) that may impact the validity of an embedded ontology. As in the case of the use of relational databases, systems using ontologies may have security vulnerabilities. Though ontologies by themselves may appear to pose no security risks, their infrastructure (e.g., reasoners, triple stores, etc.) or use in a system provides opportunities to introduce security vulnerabilities. As an example, consider vulnerabilities similar to an SQL injection attack. None of the evaluation methods or techniques discussed during the summit adequately (if at all) addressed aspects of ontology evaluation w.r.t. security. This is an open area for research.

5 Dynamics

Because of the perceived parallels between systems (including database, and software development) and ontology development stakeholders (I.e: system owners or developers) may assume that the ontology and its theory of the domain it's representing)is static over the course of its use (in a system). Ontologies can provide a system with dynamic capabilities not possible with conventional database technologies. During operations the ontology can change, conceivably even by derivation of new theories by examination of the instance data. These new possibilities of dynamic capabilities present a dilemma, for ontology and systems and software engineering, in that existing paradigms may not be applicable. How does evaluation change if the ontology (i.e., the T-box) is expected to change during operational use? Do such changes alter inferences or actions taken against the earlier (i.e., unchanged) ontological commitments? Do evaluation models and methods in use for such an ontology (such as simulation [5]) include revisiting those prior actions to ensure consistency, accuracy and correctness goals are being met both before and after the changes have been applied?

The notions of ontology evaluation(s) presented did not explicitly address dynamics of an ontology (i.e., changes to the taxonomy, predicates or relations over the course of its use) nor the evaluations thereof. However, there are efforts underway to fill this gap with tools and paradigms as suggested from the NEMO project.

6 Automation

In software engineering there are tools that automate parts of the engineering lifecycle (e.g., regression analysis). For ontology evaluation such tools do not currently exist to the extent that they provide feedback directly about an ontology (used in the system under evaluation). However there are projects underway that may automate evaluation of some criteria.

7 Scope

Ontology evaluation takes place in some context and that context imparts a scope (of applicability or validity). This context and scope directly impacts the importance of evaluation criteria in achieving the purpose of an evaluation. The appropriate values or value ranges for particular evaluation criteria may differ depending on evaluation context and its scope. The evaluation criteria, or their importance, may also depend on the lifecycle phase(s) in which they are applied. Most importantly how an ontology is expected to be used and in what operational domains an ontology is intended to be used, will impact evaluation criteria decisions.

8 Requirements

Behind, and supporting any notion ofassessment or evaluation,from a systems perspective, are requirements. How should one create specifications that adequately represent semantic requirements, or that specify conditions for the intended interpretations or models of the ontology and that can be tested or validated? How can tests be written that verify ontology requirements? While it should be expected that such requirements will be derived from more general systems or software requirements, the actual process that leads to appropriate requirements on the ontology and intended interpretations has not been studied extensively. Be that as it may, it is apparent that requirements development processes that meet the needs of ontology evaluation, as well as the needs of associated operational uses, must identify, configure and manage system level requirements with explicit descriptions of the expected use/intent, behavior, and results from the use of ontology [6].

Requirements development processes must therefore accommodate the potential complexity of systems employing ontologies, from an extrinsic or systems perspective, to be able to relate testing of extrinsic aspects to overall ontology quality, and to be able to make use of existing evaluation paradigms.

At what point should a requirements development process be expected to produce full, formal logical expressions against which the theories or expressions in an ontology can be tested for satisfaction of the semantic expectations of the ontology? An initial approach to create sound ontology requirements derivation processes may be the paradigm of patterns. From the identification of appropriate patterns for deriving ontology requirements more specific patterns or processes may be developed.

Systems and software engineering disciplines have experimented since the inception of large scale computing with methods for automating tests to evaluate requirements satisfaction. To support such efforts for ontology, requirements will need to be developed and expressed in formal fashion, equivalent or better in expressiveness from that used to specify the ontology itself. It has been demonstrated that proof techniques can be employed for three cases of requirements specification evaluations. Presumably, once the requirements themselves have been validated using these techniques, they can become specifications for the ontology that is the ultimate product of the exercise.

Without adequately expressive requirements that focus on ontology, any evaluation will fail to meet expectations. Moreover, the applicability and overall value of requirements to ontology evaluation will be related to their level of detail and the lifecycle phase at which they may be evaluated.

While requirements development and management are mature aspects of systems and software engineering disciplines, establishing reliable methods and tools for requirements refinement with a goal of economically producing correct and useful ontologies would provide leverage for ontology development and use, more securely coupling them to the associated deliverable system(s).

9 More Material

Much of this section was created based upon the excellent presentations made during the summit, which can be found at [7] and [8].

Links

[1] `http://ontolog.cim3.net/file/work/OntologySummit2013/2013-01-24_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects/A-Method-for-Development-n-Verification-of-Expressive-Ontologies--MeganKatsumi_20130124.pdf`

[2] `http://ontolog.cim3.net/file/work/OntologySummit2013/2013-02-28_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects-2/OntologySummit2013_System-Testing-in-a-Data-World--KeithSilliman_20130228.pdf`

[3] http://ontolog.cim3.net/file/work/OntologySummit2013/2013-01-24_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects/Evaluation-Context-for-OntologiesŮHansPolzer_20130124.pdf
[4] http://ontolog.cim3.net/file/work/OntologySummit2013/2013-01-24_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects/BlackBox-Testing--MaryBalboni-DougToppin-ThanhVanTran_20130124.pdf
[5] http://ontolog.cim3.net/file/work/OntologySummit2013/2013-02-28_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects-2/OntologySummit2013_Assessing-Ontologies-via-Simulation--JoaoPauloAlmeida_20130228.pdf
[6] http://ontolog.cim3.net/file/work/OntologySummit2013/2013-02-28_OntologySummit2013_OntologyEvaluation-ExtrinsicAspects-2/OntologySummit2013_From-Use-Context-to-Evaluation--AmandaVizedom_20130228.pdf
[7] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_01_24
[8] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_02_28

Track C: Building Ontologies to Meet Evaluation Criteria

Matthew West and Mike Bennett

1 Mission Statement

To investigate the state of the art in ontology development methodologies, including key achievements and key gaps that currently exist.

2 Background

There are two approaches that can be taken to assuring the quality of an ontology:

1. Measure the quality of the result against the requirements that it should meet.
2. Use a process or methodology which will ensure the quality of the resultant ontology.

If you wait to the end of ontology development to measure the quality, the costs of correction of any errors are likely to be high. Therefore using a process or methodology that builds quality into an ontology can have significant benefits. At present, however, it is unclear if there is any process or methodology that, if followed, is sufficient to guarantee the quality of a resulting ontology, and most of those that do exist are relatively informal and tend to require expert support.

A consideration in evaluating ontologies is the different scenarios in which they are used. For example, one might be used as a formal conceptual model to inform development and another might be used in an ontology based application. Both the evaluation criteria and the development methodologies employed may vary widely.

3 Objectives

1. Examine the explicit and implicit methodologies that are known to exist.
2. Understand the role that upper ontologies play in ontology development methodologies.
3. Understand the role of ontological patterns in ontology development methodologies.

4. Identify how to apply the intrinsic and extrinsic aspects of ontology evaluation identified by the other tracks, within the applicable development methodologies.
5. Identifying how to frame the applicable ontology development methodologies within the frameworks of established quality assurance regimes (such as ISO 9000 and CMMI) for industrial applications.

4 Synthesis: Input from Track C for the Summit Communiqué

4.1 Why is ontology evaluation important?

Establishing requirements (agreed between users and developers of an ontology) that an ontology needs to meet in order to meet the needs of its application means that those developing the ontology have a better chance of meeting those requirements (you can't fail to meet unstated requirements).

Confirming that an ontology meets the requirements should be part of the acceptance of an ontology in a wider systems development context. There may be several stages of development and maintenance with different levels of requirements at different stages.

When looking to reuse rather than reinvent an ontology, an evaluation of the ontology in terms of what requirements it meets, will make it easier to identify an ontology that may be appropriately reused – in whole or in part – for some other purpose.

4.2 Key considerations for ontology development methodologies

An ontology development methodology needs to provide positive answers to the following questions:

1. Is the domain represented appropriately (given the requirements of the IT system)?
2. Is the ontology human-intelligible?
3. Is the ontology maintainable?
4. Does the query/reasoning capability and performance meet the requirements of the IT system?

4.3 What is the State of the Art of Ontology Evaluation

Track C noted that for integrating ontologies, consistency was a critical property. Achieving consistency across large and potentially geographically and culturally diverse development and maintenance teams was a particular challenge in methodology development.

The development process for an ontology needs to have a number of stages, just like the data model in a traditional information systems development process. Similarly requirements need to be identified in levels too, starting with the capabilities of the overall system that the ontology is a component of, to capabilities of the ontology itself in that setting, to high level requirements, like consistency, to detailed requirements, like conforming to naming standards. The ontology development needs to go through stages to match, equivalent to conceptual, logical, and physical data model development in information systems. There are architectural decisions to be made in terms of the choices of ontological commitments the ontology needs to make and does make. There are choices of ontology language and implementation environment. There is little evidence of this in current practice, where ontology development seems to start with someone writing some OWL or Common Logic.

There is little or no integrated tool support for multilevel/multistage ontology development beyond some tools to directly support the development of ontologies at this physical level.

4.4 Future Steps

In order to improve the situation for ontology evaluation and, thus, indirectly, improve the quality of ontologies the following issues need to be addressed:

1. A better understanding of the relationships between requirements at different levels and how low level requirements support higher level requirements.
2. Ontology development methodologies that align with and recognize similar stages to information systems development with distinct conceptual, logical, and physical stages, so that ontology development does not start at the physical level with the choice of an implementation language.
3. A clearer understanding of the architecture of ontology development and the different aspects of architecture that are relevant, from ontological commitments to language choices.

5 More Material

Much of this section was created based upon the excellent presentations made during the summit, which can be found at [1] and [2].

Links

[1] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_02_07
[2] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_03_14

Track D: Software Environments for Evaluating Ontologies

Michael Denny, Ken Baclawski, and Peter Yim

1 Mission Statement

Through this track, we aim to coordinate the following: Provide a venue to bring together individuals and communities who can help define and advance the state-of-the-art in software and systems for evaluating ontologies The collection and enumeration of software environments and tools for evaluating ontologies (with emphasis on those that are open efforts and those that are publicly available)

2 Work-Products from this Track

Our approach: [1]

- Introduction of our track mission and approach
- Introduction of our approach to the survey on software support to ontology quality and fitness
- The two panel discussion sessions when we invited stewards of some exemplary ontology software tools and environments out there to share with us their work, experience and insights ...

  2013_02_14 - Thursday: Session-05: ‘Software Environments for Evaluating Ontologies - I’
  Co-chairs: Peter Yim and Michael Denny
  Panelists: Michael Gruninger, Jeanne Holm, Gavin Matthews [2]
  2013_03_21 - Thursday: Session-10: ‘Software Environments for Evaluating Ontologies - II’
  Co-chairs: MichaelDenny and PeterYim
  Panelists: Adam Pease, Till Mossakowski, Tania Tudorache, Michel Dumontier, Kingsley Idehen [3]
- The Survey on ‘Software Support for Ontology Quality and Fitness’ [4]
- Support to the Hackathon-Clinics program team at their introduction and launch [5]

3 Synthesis

The notion of tool support of quality is broader than the track's title and should include 'guidance' as well as 'evaluation' of those ontology characteristics determining an ontology's quality and fitness. Ontology tools and software environments may intentionally constrain or recommend to the user proper ontology structure and content.

Tools may contribute this 'evaluation' or 'guidance' function at different points along the ontology life cycle, and for a given characteristic, some tools may perform better in one life cycle phase than in another phase where a different tool is better suited. Generally, appreciation of the full cycle of life of an ontology is not well established within the ontology community.

There are central aspects of ontology that may not be amenable to software control or assessment. For example, the need for clear, complete, and consistent lexical definitions of ontology terms is not presently subject to software consideration beyond identifying where lexical definitions may be missing entirely. Another area of quality difficult for software determination is the semantic fitness of an ontology to its world domain (reality) or to its application domain. Software guidance may be available for the fitness of candidate ontologies for import and reuse, but not so for the novel content of a new ontology.

The design, implementation, and use requirements of an ontology may affect how quality and fitness on a particular ontology characteristic are determined, as well as interpreted and valued. Perhaps all quality and fitness assessments by software should be traceable to stated ontology requirements.

Significant new ontology evaluation tools are currently becoming available to users. Carving a link between such tools and existing IT architecture and design tools (e.g., EA and SA) remains a future possibility in order to integrate ontology into mainstream application software development within enterprise or more focused IT environments. This capability could offer a definitive means of connecting ontology quality/fitness characteristics and measures to use case and application software requirements.

Approximate lexical and structural matching of a new ontology or ontology component to the content of a repository of known ontologies may offer an effective means of identifying comparable ontology content for:

1. demonstrable coding patterns;
2. confirmation of authoring approach; and
3. identification of reuse candidates.

Links

[1] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Software_Environments_For_Evaluating_Ontologies_Synthesis
[2] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_02_14
[3] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_03_21
[4] http://ontolog-02.cim3.net/wiki/Category:OntologySummit2013_Survey
[5] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_03_28

BACnet Ontology Evalution Hackathon

Joel Bender

1 Objective and Goals

The objective of the BACnet Ontology Evaluation Hackathon was to review an OWL ontology that was built from the ASN.1 productions in the BACnet™ standard. It is early in the ontology development lifecycle.

BACnet™ is a communications protocol for Building Automation and Control Systems (BACS) developed under the auspices of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). BACnet is an American national standard, a European standard, a national standard in more than 30 countries, and an ISO global standard. The protocol is supported and maintained by ASHRAE Standing Standard Project Committee 135 (SSPC-135).

2 Background and Challenges

There are two sets of challenges; intrinsic – those related to developing an ontology that properly models what is described in the standard, and extrinsic – where the standard uses terminology that is also used in other standards, but maybe inconsistent with those other standards.

2.1 Intrinsic Challenge

BACnet specifies not just the 'on the wire' encoding and decoding of communications requests and responses, but also a rich model of 'objects' and 'properties'. Properties have restrictions on their data types and values which may be atomic values (booleans, integers, strings, etc) or structured data (lists of composite objects). Many properties are optional, and in some cases optional properties are grouped together so if some specific property exists then another property must also exist in a BACnet conferment device. Clause 21 of the standard specifies the request and response protocol data units in ASN.1 productions, and Annex C specifies the object types and properties as ASN.1, but both are inadequate for formal model analysis. However, they do provide a lexicon and naming convention which could be used to build a ontology.

2.2 Extrinsic Challenge

The building automation industry is similar to the industrial process control industry and shares may of the same basic concepts and terminology. Formally matching these concepts will facilitate software developers developing systems that can provide a holistic view of operational performance and energy use throughout a campus that may include office, research, and manufacturing buildings. Similarly, the OGC Observation and Measurement Model and the W3C Semantic Sensor Network Ontology share many of the same concepts and relationships with building automation sensor networks. There are a variety of other standards which are being incorporated into new standards under development, for example, IEC 61968/61970, IEC 61850 and the WXXM Weather Model are being incorporated into a new Facility Smart Grid Information Model (FSGIM) being built as part of the national smart grid initiative. An OWL model of the FSGIM is being developed, and BACnet would be considered one of its extrinsic stakeholders, so there is a great opportunity to bridge the two together.

3 Process

This Hackathon began with the RDF/RDFS transliteration of the structured content of a BACnet library, produced by a Python script. It was available in OWL Functional Syntax and RDF/XML that should be successfully imported into NeOn and Protege with the expectation that the same format will be acceptable to other tools. The 'hacking' consisted of a review of the development process and design considerations followed by a review of the output of analysis tools. There was a discussion of the changes that should be made to the model to re-align it to best practices and/or satisfy the recommendations of the tools.

The first challenge was picking an OWL format for exchanging the developing ontology with the team members. While all of the notations are designed to be isomorphic, so in theory the content could be presented in one notation and team members could easily translate it into a notation that they and their tools are comfortable using, not all of the freely available tools could translate easily between formats, nor could they translate back into the original published format. For most of the initial development, OWL Functional Syntax, OWL/XML, RDF/XML, and Turtle were all simultaneously produced by the script.

The next challenge was to find definitive examples of fundamental Computer Science constructs that would be considered 'best practice' and follow them as templates. While there are some examples available (like 'domain' and 'range'), it became apparent that the ontology development community does not use or prefer them, rather they describe similar concepts using other constructs (like ‘class restrictions’). It also became clear that it is unusual to describe structures that have a specific order of elements, some of which are required and some of which are optional, so while the current design is probably correct in some sense, the automated validation tools used by the team struggle with the ontology.

The challenges continued with making distinctions between labels for concepts in the standard and the named values used when marshaling the concept. One of the more difficult concepts is that of a ‘device’ and a ‘device object’ and recognizing the distinction in conversation can be difficult for someone that has not spent a great deal of time immersed in the BACnet community. The BACnet standard also defines terms like ‘network’ that do not necessarily mean the same as the same term in other standards.

4 Results

To simplify the volume of information the team needed to review, the team has settled on Turtle as a widely accepted RDF format. It is relatively simple to read and supported by a variety of tools.

The team has generally accepted the inclusion of other ontologies like RDFS and SKOS, but there is more work that needs to be done before it reaches a consensus that they are being used in the way they are intended.

The project team is still in the process of deciding how to modularize the ontology so that different semantic applications can use different modules without the need to import the entire ontology, and the module components may shift around as the team gains more experience incorporating it into systems. There are at least three distinct semantic layers in BACnet; the encoding of protocol data units (messages) as a binary string of data, the ‘objects’ that have ‘properties’ with the values that applications would exchange, and the ‘services’ which are specific messages used to exchange data according to various rules. A trivial example is ‘units’, BACnet has its own concept of ‘units’ like ‘degrees Celsius’ that are associated with measured and computed values and one of the future challenges is to map BACnet units into those used by other ontologies.

There are other layers as well, such as the concepts and terms for web services and exchanging configuration information via XML. At the time of this writing those other sections of the BACnet standard are not being considered for inclusion into the ontology.

The team continues to meet regularly and discuss the ontology and its application. The current ‘early alpha’ version of ontology, along with links to the discussion forum, is available at: `http://bacowl.sourceforge.net`.

5 Participants

Joel Bender
Steve Ray
Mike Dean
Evan Wallace
Derek Lasalle
Peter Yim
Mike Bennett
Amanda Vizedom
Astrid Duque Ramos
Bill Freeman
Bob Smith
Derek Lasalle
Francesca Quattri
Jacobus Geluk
Kevin Tyson
Max Gillmore
Maria Poveda Villalon
Mike Dean
Peter Yim
Samir Tartir
Simon Spero
Todd Schneider
Dennis Wisnosky

Links

[1] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Software_Environments_For_Evaluating_Ontologies_Synthesis
[2] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_02_14
[3] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_03_21
[4] http://ontolog-02.cim3.net/wiki/Category:OntologySummit2013_Survey
[5] http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_03_28

Evaluation of OOPS!, OQuaRE, and OntoQA for FIBO Ontologies

Mike Bennett

Hypercube

1 Objectives and Goals

This ontology clinic aimed to explore the application of ontology quality measures to ontologies produced under the Financial Industry Business Ontology (FIBO) umbrella.

In this clinic we explored the application of the OOPS! and OQuare methodologies and tools to two styles of ontology developed under the FIBO umbrella: Business Conceptual Ontologies (BCOs) which are the FIBO standards themselves; and example 'Operational Ontologies' derived from these for deployment in semantic technology applications.

We looked to establish which types of measure should be applied to each type of ontology and apply the relevant tools and techniques to these. In the case of OQuaRE, these measures will be applied in two ways: 1) application of the complete quality model; 2) application of the OQuaRE subcharacteristics and metrics relevant for FIBO evaluation, with the possibility of modifying the existing associations subcharacteristics-metrics.

From this activity made the first steps towards defining a formal quality process for the future development of formal standards under the FIBO umbrella, a set of quality assurance parameters for users who need to extend the FIBO BCO locally for their own conceptual semantic modeling, and a set of guidance notes, validation and verification techniques etc. for developers of semantic technology applications based on the FIBO standards. We evaluated to what extent OQuaRE could be a start point for this quality process.

2 Background and Challenges

2.1 FIBO

FIBO is being developed as a series of 'Business Conceptual Ontologies' (BCO) for concepts in the financial industry, that is, ontologies which represent industry terms, definitions and relationships at the level of conceptual models. Conceptual models, by definition, should not reflect application constraints. From these, we anticipate that users would derive operational ontologies for specific use cases, which would of course be subject to the relevant application constraints.

An open question in the development of FIBO is what ontology quality measures should be applied to the 'Conceptual' ontologies, and which of the established OWL modeling best practices are applicable to such an ontology. That is, which requirements of semantic technology applications should be applied to the conceptual ontologies without compromising their requirements as conceptual models.

To complicate this question further, the BCOs are intended to be presented to business domain subject matter experts for validation, and local extensions of the BCO are intended to be understood and maintained as a business domain asset. In order to support business-friendly presentation in the currently available modeling tools, some compromises have been made in the way that the OWL language is used. Some of those compromises could be reversed once there are better ways of presenting these ontologies to a business audience.

Meanwhile, we expect potential users of the standards to derive 'operational ontologies' from the conceptual ontologies, just as a conventional application developer would develop logical designs from conceptual models such as requirements catalogs. These operational ontologies must of course be subject to the quality requirements of any application (validation and/or verification of the delivered item against the stated business requirements), and since they are OWL ontologies, must also be subject to the quality constraints that are applicable to operational OWL ontologies.

2.2 OQuaRE

OQuaRE is a framework for Ontology Quality Requirements and Evaluation based on ISO/IEC 25000:2005, the standard for Software Quality Requirements and Evaluation. The complete definition of OQuaRE is available at [1] and [2].

OQuaRE defines intrinsic and extrinsic quality criteria in terms of quality sub-characteristics. OQuaRE aims to define all the elements required for ontology evaluation: evaluation support, evaluation process and metrics. The current version of OQuaRE includes, so far, the quality model and the quality metrics:

– The quality model is composed of a set of quality characteristics such as structural, functional adequacy, maintainability etc. and its associated sub-characteristics such as reliability, reusability, availability, redundancy, consistency, etc.
– The quality metrics have been taken from the state of the art in ontology, such as Depth of subsumption hierarchy, Class Richness, Tangledness etc.

3 Process

During this hackathon we covered both the evaluation of the existing ontologies as well as developing criteria for the evaluation of future ontologies.

3.1 Evaluation of FIBO Business Conceptual Ontologies

Our goal was to Identification of relevant quality metrics and aspects for FIBO Business Conceptual Ontologies. Further, we explored the use and evaluation of ontology quality tools for the evaluation of FIBO Business Conceptual Ontologies. For this purpose, we applied these measures to the the 'FIBO-Foundations' ontologies using the available tools, and considered how this can inform the formal methodology for FIBO development

3.2 Consider Criteria for Future Operational Ontologies

One important task was to Identify the relevant quality measures for a FIBO-derived Operational Ontology. For this purpose we considered how the application use case can be shown to be satisfied by a given operational ontology. In particular, we investigated whether this can be formalized in such a way that formal 'Conformance Points' can be defined which are of a suitable level of clarity and repeatability to be included in the OMG specification as formal conformance criteria. Even in cases in which these requirements and tests cannot be formalized, we considered what application guidelines can be created around these tools and techniques, to guide users of FIBO in creating robust ontology based applications which conform to their stated user requirements.

4 Results

We had practical demonstrations of all 3 tools (OOPS!, OQuaRE, and OntoQA) on real FIBO OWL ontologies, and looked at what measures the tools were showing us. We explored a couple of the metrics in depth. We looked at the OQuaRE table of quality measures and considered some changes and additions. I've tried to capture some of these in edits and additions to the table. Others will also be working asynchronously on this table.

We had a couple of strong new ideas for quality measures: Jacobus Geluk suggested having a suite of SPARQL queries that can be used as regression tests or for test-driven agile development, along with example instance data. The OntoQA tool has some tests that can be applied separately to that test data. Simon Spero suggested that the ACE plug-in for Protege can be used not only to provide business descriptions, but as a good quality measure, with a human in the loop, to test whether some of the assertions in the ontology really mean what we meant them to mean.

On OOPS!, we imported the whole set of the FIBO ontologies we had, into a single test ontology and ran the pitfall scanner on this, analyzing all the imported ontologies. We established that most of the metrics were ones we would want to apply to the FIBO Business Conceptual Ontologies, not just operational ontologies. We can determine the required values for some of these; so for instance in Foundations we want a higher level of confidence in reusability and changeability.

On OQuaRE we saw the full set of tests run against several of the FIBO Foundations ontologies. We looked at some of these in depth. These implement specific mathematical algorithms which are documented in the OQuaRE document, with cross reference to the business definitions of quality measures which they support. Some of these measures may have higher priorities in FIBO-Foundations than other ontologies, e.g. reusability.

We had a demonstration of the OntoQA tool, on the FIBO Foundations ontologies. This has metrics for individual knowledge bases as well as for the conceptual model (these may also be applied to test data). These are presented in a spreadsheet, and some of them correspond to items in the OQuaRE table.

Future versions of the tools may allow more automatic configuration of these settings. In OOPS! we would want to be able to replace their annotation measure (which looks at rdfs:label), with measures that look at SKOS definitions and the like. There are other configuration changes and new developments which would benefit FIBO, and / or the results can be filtered according to the specific requirements with what there is now.

We had an in depth discussion on some FIBO modeling parameters, including the 'Archetype' concept (similar to ontology design patterns) and how this helps with ontology quality. It should be possible to create code which would validate the application of archtypes and related ontology patterns. We also learned more about how the FIBO Archetype concept can be implemented in OWL more formally than it is at present.

In a second sessions we worked through the Google Docs document.[1] We went through the table one row at a time. For each entry we looked at the quality requirement defined for that entry, expanded this into specific measures to look at in the ontology. Then we reviewed the OQuaRE metrics in the table and added any other OQuaRE metrics that would apply.Then we added metrics from OOPS! and OntoQA which would measure some aspect of the ontology which would be relevant to the same overall quality requirement.

[1] The document is available at [3]

5 Participants

Mike Bennett
Amanda Vizedom
Astrid Duque Ramos
Bill Freeman
Bob Smith
Derek Lasalle
Francesca Quattri
Jacobus Geluk
Kevin Tyson
Max Gillmore
Maria Poveda Villalon
Mike Dean
Peter Yim
Samir Tartir
Simon Spero
Todd Schneider
Dennis Wisnosky

Links

[1] http://miuras.inf.um.es/evaluation/oquare/
[2] http://miuras.inf.um.es/oquarewiki/
[3] https://docs.google.com/document/d/1ErbZV0IFj890lHFcnygsw6n93d xub1AamOu9oBnHdOo/edit

The General Ontology Evaluation Framework (GOEF) & the I-Choose Use Case

Joanne S. Luciano, Nicolau DePaula, Djoko Sigit Sayogo, and James Michaelis

1 Objective and Goals

We used the 2013 Ontology Summit as an opportunity to present and advance the development of the General Ontology Evaluation Framework (GOEF). We used an I-Choose use case and ontology to study the GOEF approach in practice. While the I-Choose use case is real, the development of GOEF is early stage and exploratory. The extent to which this work was carried out in the context of GOEF should be thought of as proof-of-concept.

2 Background and Challenges

In current practice, use case driven ontology evaluation is managed through direct inspection by subject matter experts. This is a time-consuming effort, that requires individual review of potentially multiple ontologies. Thus, we ask, what if we could develop a system which could take in a use case formalism, and give recommendations for ontologies to use? The motivation for the development of GOEF was determine if an ontology or part of an ontology could be reused for a different purpose.

The goal of GOEF is to enable objective evaluation of an ontology with respect to a use case. The GOEF approach can be used as an modular incremental ontology development methodology and for existing ontology evaluation. When used as a development approach, GOEF facilitates ontology design, modular construction, and development management because evaluation is built into the development process at every step. The GOEF approach consists of two stages: recasting of a use case into three distinct levels of requirements and components, and evaluation of components using objective metrics at each level. The objective metrics are intrinsic and extrinsic, many previously defined, others yet to be articulated and will need to be defined. Metrics are applied to, but independent of the object of measurement, for instance, metric units (e.g. liters) are independent of the amount (e.g. amount of wine or beer or milk or water) being measured in the glass.

One objective of I-Choose project is to develop an interoperable data framework to provide consumers with a wide range of information about how, where, and by whom products are manufactured and brought to market. More specifically, the I-Choose ontology architecture is intended to enable consumer advocates the capability to retrieve and verify information acquired from social and environmental certification procedures, such as inspections undertaken to acquire FairTrade and Organic certifications. The I-Choose network and ontology team decided to focus the first application ontology on the FairTrade coffee certification. This preliminary ontology would support a prototype application to be developed and tested with data (actual or artificial). Ultimately, the I-Choose project is interested in developing a data standard for this sort of inspection/certification process.

3 Process

During our participation in the Ontology Summit 2013, our goals centered on the development of a case study example that would enable us to explore the application of the GOEF in the context of I-Choose. Here, our contributions were twofold: (i) we identified a relevant and descriptive use case for an ontology application, and (ii) we applied the GOEF methodology to identify the corresponding functional semantic components and evaluation metrics from the use case.

3.1 Identifying Use Cases in I-Choose

I-Choose aims to develop a proof-of-concept approach for an interoperable data architecture that supports the provision of trusted information about unobservable product attributes. The current focus is on voluntary sustainable certification for coffee (Jarman et al., 2011; Luna-Reyes et al., In Press). The major component of the envisioned I-Choose system is a set of ontology-based data standards that will enable users to extract data and information from voluntary sustainable certifications for the coffee supply chain. The envisioned power users of I-Choose are the online consumer advocates. I-Choose is envisioned to enable these consumer advocates with the ability to trace and extract information from the supply chain in order that they may deliver trusted information about sustainable coffee products to consumers. One use case of the proposed ontology is to verify that a coffee product is not tainted with child labor. A consumer advocate may want to trace the certification information in order to verify that the the producer is in compliance with the “No Child Labor policy.” This scenario envisions that a consumer advocate has the ability to access the inspection reports within the database of a Certification Body. In this scenario, the consumer advocate is more interested in understanding the robustness of the Child

Labor policy from particular certification schemes. In this case, the consumer advocate wants to extract the list of criteria governing "No Child Labor" from a particular certification scheme. The consumer advocate could send a query to the certification body, such as: "What are the core indicators for protection of child labor for this label in conjunction the ILO minimum age convention?"

3.2 Application of GOEF Methodology

In general, although not explicitly stated as such, use case descriptions are expected to contain important contextual information that is needed to evaluate the applicability of the ontology for an intended use case.

The first stage of GOEF - Recast a use case into its contextual components and levels. This consists of conducting the following sub-tasks:

- **Establish functional objectives:** These represents the top level of the use case - which we view as the the function or the intended use (e.g. for search, for data integration, for certification). Additionally, functional objectives represent the primary characteristics that define the classification of the domain of the ontology (e.g. an airplane, an elephant, a financial instrument).
- **Establish requirements compliance specification):** This represents the quality or standard that has to be met by the application (e.g. legal, interoperability, compliance, etc.). These are extrinsic to the function (e.g. business requirements, ISO Standards, MIL specifications, or Minimum Information content standards).
- **Establish needed semantic components:** Here, needed ontology fragments are identified which fulfill the functional objectives and meet the level of the requirements specification standard. (e.g., for an aircraft specification, representations of wings, engine, wheels would be needed). In complex systems, these may be composite; many "standard" components reused (e.g., nuts and bolts used in doors and wheels)). This is where modularity comes in enabling individual components to be evaluated and reused.

Following the identification of functional components, the second stage of GOEF involves evaluating these components using objective metrics. These metrics fall under the categories of correctness, completeness and utility. Additional categories such as extensibility could also be considered. The I-Choose ontology can be evaluated for the I-Choose certification use case using these three categories. For instance, correctness can be measured using two metrics. First, by matching information provided in the ontology to a general accepted definition. In this case, the general accepted definitions for child labor are International Labour Organization (ILO) Convention No. 138 and No. 182. Second, by evaluating the syntactical validation of the ontology and logical consistency. Completeness can be measured by evaluating if all the child labor criteria and necessary characteristics are covered in the ontology. For example, testing the ontology to distinguish between compliant versus non-compliant to the criteria. Utility can be measured from the capability of the ontology to satisfy the questions from the consumer advocate through a query interface of sample data.

For the purpose of our involvement in this summit, one of our main objectives was to identify corresponding functional components of our sample use case, as well as appropriate evaluation metrics.

4 Results

Following the identification of a use case from I-Choose, both stages of the GOEF approach were applied. Our approach to recast the use case into functional components yielded a record of the following form:

Function: Enable retrieval of specific compliance criteria from a specific inspection process of a particular product.

Requirement Compliance Specfication: Initial system: Satisfy consensus user criteria pre-determined by survey research.

Semantic Components:

- *Certification Criteria*: (a) Children Below 15, (b) Under 18 dangerous work, (c) Preventive measures ensure safety
- *Certification Body*: (a) Flo-Cert
- *Certification Standard Setter Organization*: (a) FairTrade International, (b) USDA Organic, (c) ISO 65
- *Product*: (a) Coffee

Based on this extraction of functional components, we were in turn able to identify the following objective metrics for evaluating the corresponding ontology.

Correctness:

- General logical/syntactical validation
- Are the right terms used (compliance criteria vs. code of conduct vs. standards)
- Match information provided in the ontology to general accepted definition provided by ILO (e.g. age of 15 is accepted standard)

Completeness:

- All child work criteria, and necessary characteristics included
- Ability of ontology to distinguish compliant vs. non-compliant criteria

Utility:

- Consumer/Consumer Advocate Questions – Satisfied

5 Discussions and Next Steps

This summit allowed developers of the I-Choose ontology and developers of the Generalized Ontology Evaluation Framework (GOEF) to interact with ontology experts in a discussion of ontology evaluation. We received a number of helpful comments from the Ontology Summit Expert Panel: Peter Yim, Mike Dean, Leo Obrst, and Ken Baclawski. The panel presented a examples of relevant developments in ontology creation and evaluation, provided specific references for the next steps we envisioned, and made positive contributions to the future of both the GOEF and I-Choose projects. We appreciated the support of the Ontology Summit and the Expert Panel and their insightful remarks, the highlights of which we summarize below.

Leo Obrst carefully observed the propositions of the GOEF framework and elaborated on the differences between *intrinsic* and *extrinsic* evaluation. He commented that intrinsic evaluations often refer to *primary domain components* and consist of evaluating quantifiable characteristics of these components (e.g. average, density, etc.). Extrinsic evaluations require moving from very basic properties of a domain to, ultimately, particular applications. Obrst added that these applications may serve one or multiple use cases. As such, he continued, the combination of intrinsic and extrinsic evaluations, that GOEF proposes to standardize, must encompass this continuum, where first intrinsic evaluations have to be made according to basic properties of a domain and where extrinsic evaluations serve to gauge the success of particular applications. Thus, if a generalized ontology framework is to succeed it has to consider the different characteristics that ontologies possess dependent on the continuum where they are found. We agree.

One of the early observations in the GOEF approach of ontology evaluation in accordance with a use case is that a formalized approach to use case design must be developed. The authors had done some exploration in this area and had personal communication with Alistair Cockburn, author of Writing Effective Use Cases and author of the template we use, however we had not yet started our work in this area. We were therefore grateful to Ken Baclawski for bringing to our attention his own work in use case formalization. Ken presented to us an ontology that formalizes use case descriptions, explicitly identifying the actors, systems and activities found in use case diagrams that is available in OWL and UML. This ontology of use cases could be used in GOEF for the representation of the use case that is needed for the evaluation. As Ken pointed out, the next step is to build a tool which compares objects in the formalized use case to those in a corresponding ontology.

Peter Yim and Mike Dean made further observations on the current work of the *ontology of ontology evaluation*in which Mike Dean is a participant. With insights from GOEF and the ontology of use cases, these three projects seemed to cover most of the steps necessary to achieve a formalized framework for evaluating ontologies based on particular use cases. Peter Yim also pointed to the importance of I-Choose, as it develops an ontology of sustainable certification, to become involved in the movement for ontology-based standards.

More documentation on the Hackathon is available at [1].

6 Participants

Joanne Luciano, Djoko Sigit Sayogo, James Michaelis, Ken Baclawski, Leo Obrst, Mike Dean, Nicolau DePaula, and Peter Yim.

Links

[1] `http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Hackathon_Clinics_GOEF_IChoose`

Ontology of Ontology Evaluation

Amanda Vizedom

1 Objective and Goals

The goal of the Ontology of Ontology Evaluation Hackathon was to develop a formal ontology representing ontology evaluation elements, factors, relationships, processes, etc., as they have emerged from summit discussion.

2 Background and Challenges

The 2013 Ontology Summit provided a rich range of perspectives on ontology evaluation. Some of these perspectives are fairly abstract, some are encoded in methods and practices, and some are encoded in tools. Critical interaction via summit sessions and discussion has resulted in greater sharing of knowledge and in richer understandings of ontology evaluation at multiple elements. The purpose of this Hackathon was twofold: to use the process of ontological analysis to sort and clarify the many aspects of ontology evaluation raised during the summit; and to capture this knowledge in a precise, reusable and machine-actionable form. Such capture was intended to directly support measurement science and tools for ontology evaluation.

The primary challenge for this Hackathon was the limited amount of time available relative to the objective. In light of this challenge, the scope of work for the Hackathon was restricted to the development of a consensus conceptual model, designed to enable subsequent rigorous formalization.

3 Process

Participants worked together for one full day, and a subset of participants worked together for a further half-day. Collaboration was accomplished via teleconference, online chat, and shared documents supporting simultaneous editing, with the work of each contributor visible in real-time to others. The hackathon proper ended at the end of the second, half-day session. Short, priority follow-up tasks (mostly related to presentation of results back to the Summit community) were taken on at that time. Additional follow-on actions were suggested and discussed, with some participant commitment, for work beyond the duration of the hackathon or Summit.

The first period of work consisted of several hours of collaborative requirements gathering. This process involved all participants finding and placing in a single text document: (a) expressions of likely use cases; (b) documentation of the coverage and structure of existing models of ontology evaluation (including those implicit or explicit in evaluation tools or methodologies); (c) aspects of ontology evaluation raised during the Ontology Summit to date, with at least moderate community support for their significance (including those presented by Summit panelists; and (d) any additional participant notes regarding coverage, structure, emphasis, or other requirements for an effective model of ontology evaluation. This effort resulted in a fifty-two page document containing some redundancy but including a good representation of the considerations, and kinds of considerations, that are relevant to the meaningful, actionable evaluation of an ontology.

The second period of work consisted in all participants working together on a single, simplified graphic representation of principle aspects of ontology evaluation, or groups thereof, that must be included. This effort involved considerable discussion of the relative importance, factoring, grouping, and relationships between items raised in the raw requirements-gathering document. Since the hackathon participants varied signficantly in the perspectives from which they approached ontologies and ontology evaluation, this process also included a great deal of helpful clarification. Resulting descriptions and definitions of model elements were captured in annotations to the draft graphic model or in separate notes. This annotated graphic draft document was designated the *High-Level Conceptual Model*.

The third period of work consisted in the identification of clusters of work needed to fill out areas of the High-Level Conceptual Model. Driven partly by limitations in collaboration tools being used, the group decided to represent these clusters initially as separate graphic models, with the documented interpretation that a graph elements with the same name, whether appearing multiply in a single graph or in separate graph, represented the same thing. Individual clusters were then taken on by individual participants for rapid prototyping.

The resulting graphs were produced in simple graphic files within the same shared document store, and reviewed by the group as a whole. This review took place during the end of the first work day and the beginning of the second work day. The resulting graphs, taken with accompanying textual annotations, represented a consensus of the group.

The final period of work, taking place during the final period of the second half-day, consisted of discussion of the status of the existing graphs, how they should be connected and modeled, and what work remained. Follow-on tasks, both immediate and for later, were identified. Immediate follow-on tasks were taken on some participants.

4 Results

The results of the Hackathon are available at [1]. As noted in the main section of the report, the scope of discussion within the Ontology Summit exceeded that which could be effectively discussed within the Summit Communique. Participants in this Hackathon agreed that an adequate, actionable model of ontology evaluation should enable the documentation, tracking, and even reasoning over the aspects of evaluation that should be or have been conducted for a particular ontology and use case. Participants also agreed that such capability required inclusion of the broader range of considerations relevant to the evaluation of ontologies. The High-Level Conceptual Model reflects this consensus, giving significant representation to such factors as the operational environment and business requirements applicable in a specific situation, and the relevance of specific aspects of evaluation to such requirements. The High-Level Conceptual Model also reflects the many ways of grouping characteristics of ontologies, and the relationship between such characteristics, methods of evaluation, and metrics. In addition, the High-Level Conceptual Model is explicitly intended to support the representation of how particular evaluation tools and methods approach ontology evaluation, and what they do and do not cover. To that end, a graphic model was also produced representing the evaluation framework used by OQuare and how it fits into the general model. In all of this, the construction of the High-Level Conceptual Model, and the clarification and discussion that went into it, were enormously productive.

Several follow-on tasks were identified that have yet to be completed. The first, a unified graphical representation of the High-Level Conceptual Model and all sub-models produced, has been hampered by limitations in tools available to the taskee, such that a multi-dimensional graphical model with attached annotations and sufficient in-built precision to support formalization work can be effectively produced, without pre-fitting this model to the requirements of a specific representation language. A descriptive English version of the unified model was also taken on, but awaits the unified graphic conceptual model and/or sufficient available time on the part of the taskee. An OWL model has been drafted but will need revisiting; the model was created prior to completion of the unified conceptual model, for one thing, and also makes to attempt to capture the full annotations or complexity of relations represented in the conceptual model. A Common Logic model awaits completion of the unified graphical model.

5 Participants

Amanda Vizedom	Bob Smith	Yu Lin
Ali Hashemi	Pavithra Kenjige	Astrid DuqueRamois
Joel Bender	Peter Yim	Doug Foxvog
David Whitten	Mike Dean	Francesca Quattri

Links

[1] https://drive.google.com/folderview?id=0B_5lZBgIG6LRQnZSdUFWX2FnYTg&usp=sharing

ISO 15926 Reference Data Validation

Victor Agroskin

1 Objective and Goals

Our ontology clinic was aimed at the evaluation of publicly available ISO 15926 reference data, viewing it as an ontology for the engineering domain. We've planned to check compliance to upper ontology constraints, diagnose problems in reference data, evaluate ease of understanding and use of existing data, and make suggestions for ontology improvement. Another goal was to apply formal ontology quality metrics for data in question. Rules and algorithms were planned to support both generic verification tests and specialized checks and quality metrics designed specifically for ISO 15926 reference data.

2 ISO 15926

The ISO 15926 is a standard for engineering data integration, sharing, exchange, and hand-over. The standard defines a generic data model as an upper ontology for an engineering domain. Extensive Reference Data Libraries (ontology data) for process plants are developed by community of users, which includes equipment manufacturers, engineering companies and owner/operator companies in oil and gas, nuclear power, petrochemical industries, and others. Standard development is jointly managed as JORD project by two industry associations: POSC Caesar Association and FIATECH.

Upper ontology of the ISO 15926 is publicly available as an OWL representation of ISO 15926-2 at [1]. See [2] for more details. The biggest available Reference Data Library of POSC Caesar Association (PCA RDL) is maintained as a reference data service with human browser access and query page [3], and the SPARQL endpoint [4]. Snapshot file with full content of an endpoint is published for download [5]. Some resources resources for study of ISO 15926 are available at [6], [7] (see self-education guide at [8]).

3 Challenges

Publicly available ISO 15926 reference data is used widely and enters the stage of fast growth with the growth of awareness of the standard and increased trust in its maintainers. At the same time systematic approach to verification and quality assessment is only planned by reference data owners. We've thought that provision of basic test and measures will help the community of users and tool developers.

The following efforts were planned for the clinic:

a) Verification of upper ontology (currently distributed between 6 OWL files). This representation is heavily criticized and a new effort to do a standard representation in OWL 2 is currently suggested to ISO.

b) Check of reference data compliance to the standard and specifically to upper ontology restrictions. There are many requirements to reference data library content coming from ontology restrictions: abstract classes and disjoint classes, mandatory relationships (properties), membership relations between classes of classes and classes, etc.

c) Filling gaps in entities and relations. We've planned to base this effort on methods outlined in [9]

d) Calculation of ontology quality metrics for ISO 15926 upper ontology and Reference Data Library in various software environments or by standalone tools. Existing methodologies (like OntoClean, OQuaRE or OntoQA) were studied for this goal.

Non-standard representation of ISO 15926 reference data (RDF/OWL representation made in unconventional ways) was an additional challenge. Non-standard representation and huge size of data (almost 3 ml. triples) made useless majority of general-purpose ontology analysis tools and software packages without significant customization.

4 Place, Participants and Tools

Ontology clinic took place on Sat 2013.03.30 in 4 hour combined virtual and real session and later in the day in 2 hour virtual session. Real session gathered 5 participants in Moscow, more people from Moscow, St. Petersburg, Surgut, Kiev, Zurich connected online. In "open webcast" hour clinic's team was joined by experts from UK, Spain, USA.

Victor Agroskin	Igor Katrichek	Astrid Duque Ramos
Alex Ivanov	Martin Davtyan	Francesca Quattri
Anatoly Levenchuk	Peter Yim	
David Leal	Shishkin Dmitri	

Implementation work was carried out in the environment of .15926 Editor [10]. All scripting was in Python, scripts were designed for execution in the Python console of .15926 Editor software using its API for access to RDF data.

Other tools were tried, debugged and improved in the process but had not delivered reportable results.

5 Process

Of the planned efforts listed above only two tasks delivered real results: reference data compliance checks and ontology quality metrics' assessment.

Basic taxonomy tree of the reference data library was checked for the presence of unconnected components and specialization loops. Mandatory relationships were checked for classes representing UOM ontology.

Various mistakes were identified in relations between classes of individuals, classes of relationships and classes of abstract objects. Class membership restrictions implied by upper ontology were verified for a significant part of reference data and problems in upper ontology representation were diagnosed for the remaining part.

The data was checked for missing concept definitions.

The team worked on OQuaRE quality metrics' [11] adaptation for PCA reference data library. Metric calculation scripts were designed for 5 metrics, characterizing the structure of ontology (mean specialization path lengths, mean number of properties and relationships per class, length of the largest path, mean number of ancestor and child classes).

6 Results

Clinic activities were instrumental in bringing together tool developers and users, providing an opportunity to demonstrate software usage patterns. Strong and weak sides of software tools in ontology exploration environment were identified which will guide further development of .15926 and other software.

Verification scripts and test results were reported to ISO 15926 community and to reference data library maintainers on community portal forum [12].

Ontology quality metric measurements were submitted to the Ontology Summit community mailing list [13]. Further quality comparison of PCA reference data library with other ontologies can be carried out once comparable data is collected in OQuaRE project.

Links

[1] `http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003`
[2] `https://www.posccaesar.org/wiki/ISO15926inOWL`
[3] `http://posccaesar.org/endpoint/`

[4] http://posccaesar.org/endpoint/sparql
[5] http://rds.posccaesar.org/downloads/PCA-RDL.owl.zip
[6] http://www.infowebml.ws/,http://www.15926.info/
[7] http://www.15926.org/
[8] http://levenchuk.com/2012/10/01/iso-15926-self-education-sequence/
[9] http://ontolog.cim3.net/file/work/OntologySummit2013/2013-03-07_OntologySummit2013_OntologyEvaluation-IntrinsicAspects-2/OntologySummit2013_debugging_is-a_structure--PatrickLambrix_20130307.pdf
[10] http://techinvestlab.ru/dot15926Editor
[11] http://miuras.inf.um.es/oquarewiki/index.php5/Quality_metrics
[12] http://15926.org/viewtopic.php?f=5&t=154
[13] http://ontolog.cim3.net/forum/ontology-summit/2013-04/msg00038.html

Ontohub-OOR-OOPS! Integration

Till Mossakowski

1 Objective and Goals

The goal of this event was to combine elements of Ontohub, OOR, and OOPS!, and create a web-based repository storing the feedback provided by OOPS! for many OWL ontologies, making it web-searchable and versioned. This means that the evolution of ontologies according to the feedback can be traced.

The expected outcomes were

- a webservices API that allows OOPS! to integrate with Ontohub, OOR or possible other software tools and environments.
- functionality provided online at Ontohub.org
- useful feedback to improve OOPS!.

2 Background and Challenges

The Open Ontology Repository [1] is a joint effort in providing an ontology repository that significantly goes beyond the BioPortal repository [2] in being more general (more domains and ontology languages), providing more services and being based on a decentralized architecture decoupled into several services [3]. Ontohub.org [4] is a distributed heterogeneous ontology repository that aims at implementing the open ontology repository architecture. OOPS! is a web service detecting some of the most common pitfalls appearing when developing ontologies [5].

3 Process

The API that OOPS! provides [6] was used for an ad-hoc integration with OntoIOp. The OOPS! output is integrated into the Ontohub.org display of classes and properties. The integration was implemented by a agile development approach, with multiple concept-implementation-feedback loops during the day.

Further, we discussed a general API (in the OOR context) for services like OOPS!, and have this as an extension of the BioPortal resp. OOR API.

4 Results

The Hackathon had two main results

1. OOPS! has been integrated into Ontohub. For OWL ontologies, Ontohub now displays an "Evaluate" button, with which OOPS! can be called. The OOPS! results are then displayed within Ontohub, along with the classes and object properties.
2. An OOR / Ontohub API was developed. The OOR API is currently a REST API and is the same as BioPortal, see [7]. Ontohub needs a much richer and sometimes also different API. This is because Ontohub not only supports OWL ontologies, but also ontologies written in other languages, e.g. Common Logic.
   The OOR / Ontohub API covers a wide range functionalities including parsing and static analysis, persistence, both local and distributed inferences, federation, and evaluation. Figure 1 provides an overview. The API is described in detail in [8]. Currently, a more formal specification of the API is developed in the OMG interface description language (IDL), see [9]. Later, the API will be implemented as part of Ontohub.

5 Participants

Till Mossakowski	Christian Clausen	Henning Mueller
Oliver Kutz	Timo Kohorst	Francesca Quattri
KenBaclawski	Danviel Couto Vale	Peter Yim
Maria Poveda Villalon	Julian Kornberger	

Links

[1] `http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository`
[2] `http://bioportal.bioontology.org`
[3] `http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository_Architecture/Candidate03`
[4] `http://www.ontohub.org`
[5] `www.oeg-upm.net/oops`
[6] `http://oops-ws.oeg-upm.net/`
[7] `http://www.bioontology.org/wiki/index.php/NCBO_REST_services`
[8] `https://docs.google.com/document/d/1ZTSCchPcjCPnVfKjqS1HLyTeEqh7svJghHiJhnOd5x4/edit?usp=sharing`
[9] `https://github.com/ontohub/OOR_Ontohub_API/tree/master/src`

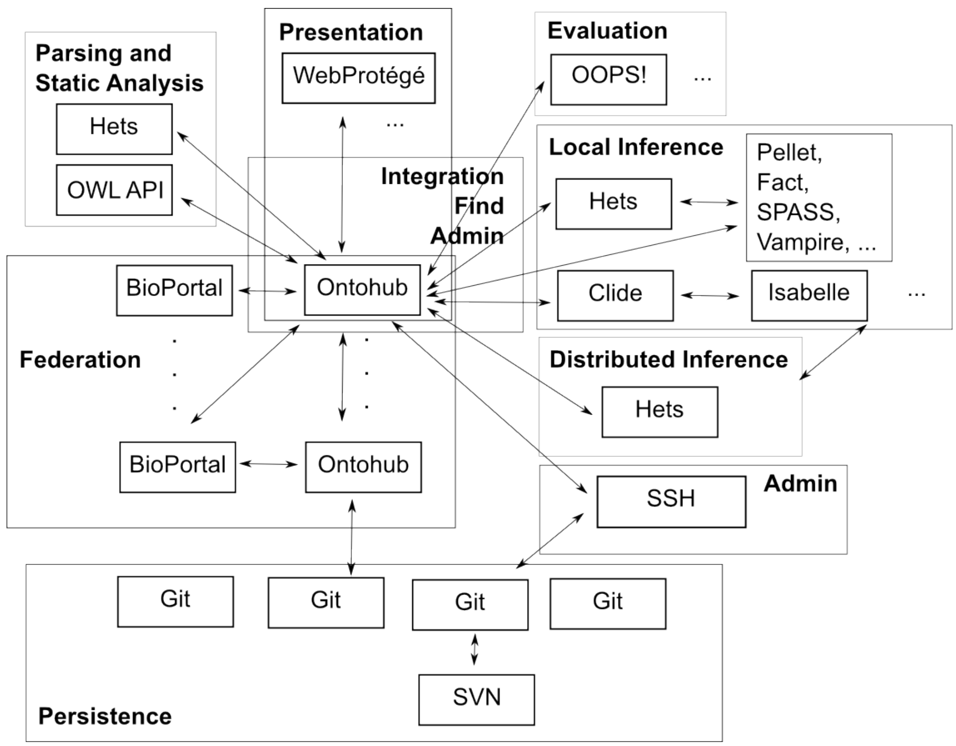

Fig. 1. OOR / Ontohub API Overview

Section 2
Toward Ontology Evaluation across the Life Cycle

The Ontology Summit Communiqué 2013

Lead Editors
Fabian Neuhaus and Amanda Vizedom

Co-Editors
Ken Baclawski, Mike Bennett, Mike Dean, Michael Denny, Michael Gruninger, Ali Hashemi, Terry Longstreth, Leo Obrst, Steve Ray, Ram Sriram, Todd Schneider, Marcela Vegetti, Matthew West, and Peter Yim

Executive Summary

Problem

Currently, there is no agreed on methodology for development of ontologies, and there is no consensus on how ontologies should be evaluated. Consequently, evaluation techniques and tools are not widely utilized in the development of ontologies. This can lead to ontologies of poor quality and is an obstacle to the successful deployment of ontologies as a technology.

Approach

The goal of the Ontology Summit 2013 was to create guidance for ontology developers and users on how to evaluate ontologies. Over a period of four months a variety of approaches were discussed by participants, who represented a broad spectrum of ontology, software, and system developers and users. We explored how established best practices in systems engineering and in software engineering can be utilized in ontology development.

Results

This document focuses on the evaluation of five aspects of the quality of ontologies: intelligibility, fidelity, craftsmanship, fitness, and deployability. A model for the ontology life cycle is presented, and evaluation criteria are presented in the context of the phases of the life cycle. We discuss the availability of tools and the document ends with observations and recommendations. Given the current level of maturity of ontology as an engineering discipline, any results on how to best build and evaluate ontologies have to be considered as preliminary. However, the results achieved a broad consensus across the range of backgrounds, application foci, specialties and experience found in the Ontology Summit community.

Recommendations

For more reliable success in ontology development and use, ontology evaluation should be incorporated across all phases of the ontology life cycle. Evaluation should be conducted against carefully identified requirements; these requirements depend on the intended use of the ontology and its operational environment. For this reason, we recommend the development of integrated ontology development and management environments that support the tracking of requirements for, and the evaluation of, ontologies across all phases of their development and use.

1 Purpose of this Document

The purpose of this document is to advance the understanding and adoption of ontology evaluation practices. Our focus is on the critical relationships between usage requirements, the life cycle of an ontology, evaluation, and the quality of the result.

This document is rooted in the 2013 Ontology Summit. Over four months, Summit participants prepared and presented materials, shared references, suggested resources, discussed issues and materials by email list, and met virtually each week for presentations and discussions. This Summit had the focal topic "Ontology Evaluation across the Ontology Lifecycle." This document represents a synthesis of a subset of ideas presented, discussed, and developed over the course of these four months, and reflects the contributions of the Summit's participants and the consensus of the Summit community.

The intended audience for this document is, first and foremost, anyone who is developing or using ontologies currently, or who is on the cusp of doing so. We believe that the adoption of ontology evaluation as presented here has the potential to greatly improve the effectiveness of ontology development and use, and to make these activities more successful. Thus, our primary audience is the developers of ontologies and ontology-based systems. A secondary audience for this document is the community of software, systems, and quality assurance engineers. When ontologies are used in information systems, success depends in part on incorporation of ontology evaluation (and related activities) into the engineering practices applied to those systems and their components.

2 Introduction

Ontologies are human-intelligible and machine-interpretable representations of some portions and aspects of a domain. Since an ontology contains terms and their definitions, it enables the standardization of a terminology across a community or enterprise; thus, ontologies can be used as a type of glossary. Since ontologies can capture key concepts and their relationships in a machine-interpretable form, they are similar to domain models in systems and software engineering. And since ontologies can be populated with or linked to instance data to create knowledge bases, and deployed as parts of information systems for query answering, ontologies resemble databases from an operational perspective.

This flexibility of ontologies is a major advantage of the technology. However, flexibility also contributes to the challenge of evaluating ontologies. *Ontology evaluation* consists of gathering information about some properties of an ontology, comparing the results with a set of requirements, and assessing the ontology's suitability for some specified purpose. Some properties of an ontology can be measured independently of usage; others involve relationships between an ontology and its intended domain, environment, or usage-specific activity, and thus can only be measured with reference to some usage context. The variety of the potential uses of ontologies means that there is no single list of relevant properties of ontologies and no single list of requirements. Therefore, there is no single, universally-applicable approach to evaluating ontologies.

However, we can identify some kinds of evaluation that are generally needed. To determine the quality of an ontology, we need to evaluate the ontology as a domain model for human consumption, the ontology as a domain model for machine consumption, and the ontology as deployed software that is part of a larger system. In this document, we focus on five high-level characteristics:[1]

1. Can humans understand the ontology correctly? (*Intelligibility*)
2. Does the ontology accurately represent its domain? (*Fidelity*)
3. Is the ontology well-built and are design decisions followed consistently? (*Craftsmanship*)
4. Does the representation of the domain fit the requirements for its intended use? (*Fitness*)
5. Does the deployed ontology meet the requirements of the information system of which it is part? (*Deployability*)

For *intelligibility*, it is not sufficient that ontologists can understand the content of the ontology; all intended users need to be able understand the intended interpretation of the ontology elements (e.g., individuals, classes, relationships) that are relevant to their use-case. Intelligibility does not require that users view and understand the ontology directly. To enable intelligibility, the documentation of the ontology needs to be tailored to the different kinds of users. This may require multiple annotations of an element of the ontology suitable for different audiences (e.g., to accommodate language localization and polysemous use of terms across domains). Intelligibility is particularly important for ontologies that are used directly by humans as a controlled dictionary. But it is also desirable for ontologies that are intended to be used "under the hood" of an information system, because these ontologies need to be maintained and reviewed by people other than the original ontology developers. *Fidelity* is about whether the ontology represents the domain correctly, both in the axioms and in the annotations that document the ontology for humans. *Craftsmanship* is concerned with the question whether the ontology is built carefully; this covers aspects ranging

[1] There are different approaches to clustering aspects of ontologies to be evaluated. For example, in the OQuaRE [1] framework, characteristics are broken down into sub-characteristics, which are linked to metrics. For more on OQuaRE and other approaches, see the Ontology Characteristics and Groupings reference collection. [2]

from the syntactic correctness of the ontology to the question whether a philosophical choice (e.g., four-dimensionalism) has been implemented consistently. Both *fitness* and *deployability* are dependent on requirements for the intended usage. These requirements might be derived from the operational environment, in the case of an ontology that is deployed as part of an information system; alternatively, they may be derived from the goals for the knowledge representation project, if the ontology is deployed as a standalone reference ontology. While both characteristics are about meeting operational requirements, they are concerned with different aspects of the ontology: fitness is about the ontology as a domain model, deployability is about the ontology as a piece of software.[2]

Since fitness and deployability are evaluated with respect to requirements for the intended use-case, a comprehensive look at ontology evaluation needs to consider how the requirements for the ontology derive from the requirements of the system that the ontology is a part of. Furthermore, although "ontology evaluation" can be understood as the evaluation of a finished product, we consider ontology evaluation as an ongoing process during the life of an ontology. For these reasons, we embrace a broad view of ontology evaluation and discuss it in the context of expected usage and the various activities during ontology development and maintenance. In the next section we present a high-level breakdown of these activities, organized as goal-oriented phases in the ontology life cycle. Afterward we identify, for each phase, some of the activities that occur during that phase, its outputs, what should be evaluated at the stage. The document concludes with some observations about the current tool support for ontology evaluation, and recommendations for future work.

3 An Ontology Life Cycle Model

The life of any given ontology consists of various types of activities in which the ontology is being conceived, specified, developed, adapted, deployed, used, and maintained. Whether these activities occur in a sequence or in parallel, and whether certain kinds of activities (e.g., requirements development) only happen once during the life of an ontology or are cycled through repeatedly depends partially on how the development process is managed. Furthermore, as discussed above, ontologies are used for diverse purposes; thus, there are certain kinds of activities (e.g., the adaptation of the ontology to improve computational performance of automatic reasoning) that are part of the development of some ontologies and not of others. For these reasons, there is no single ontology life cycle

[2] Fidelity, craftsmanship, and fitness are discussed in more detail in the section on ontology development on page 63.

with a fixed sequences of phases. Any ontology life cycle model presents a simplified view that abstracts from the differences between the ways ontologies are developed, deployed, and used.

In spite of presenting a simplified view, an *ontology life cycle* model is useful because it highlights recognizable, recurring patterns that are common across ontologies. The identification of *phases* allows the clustering of activities around goals, inputs, and outputs of recognizable types. Furthermore, a life cycle model emphasizes that some of the phases are interdependent, because the effectiveness of certain activities depends on the outputs of others. For example, effective ontology development depends on the existence of identified ontology requirements; if requirements identification has been omitted or done poorly, ontology development is very unlikely to result in useful outputs. While the development process may vary considerably between two ontologies, there are invariant dependencies between phases.

Figure 1 presents the phases of an ontology life cycle model. Typically, an ontology will go through each of these phases more than once during its life. In the following sections, each of these phases is discussed in more detail, including input, outputs and relationships between the phases. As the figure illustrates, evaluation should happen throughout the ontology life cycle, varying in focus, process, and intensity according to phase-appropriate requirements. In the following sections each of the phases is linked to phase-appropriate evaluation activities. The evaluation provides information about the degree to which requirements of the life cycle phase are being met. Requirements identification will be discussed in greater detail in the next section.

This model applies to ontologies regardless of whether their use involves significant machine processing of ontology content. Because the systems into which ontologies are incorporated are information systems in a broad sense: systems of people, processes, hardware, software, and data that process information and make decisions.[3] For example, consider an ontology which is used by humans to curate documents. In this case the information system includes the ontology, the curators, and the tools they use to browse the ontology and to annotate the documents. The success of the whole system depends on the interaction of the ontology with both the other software components and the curators. For example, the whole system will be impaired if the ontology contains information that the browser cannot display properly, or if the definitions in the ontology are so ambiguous that different curators are not able to use the terms consistently in the curation process. Thus, the ontology needs to be evaluated for both deployability and intelligibility. This example illustrates that even if

[3] See the Broad view of Information Systems reference collection for more on this understanding. [3]

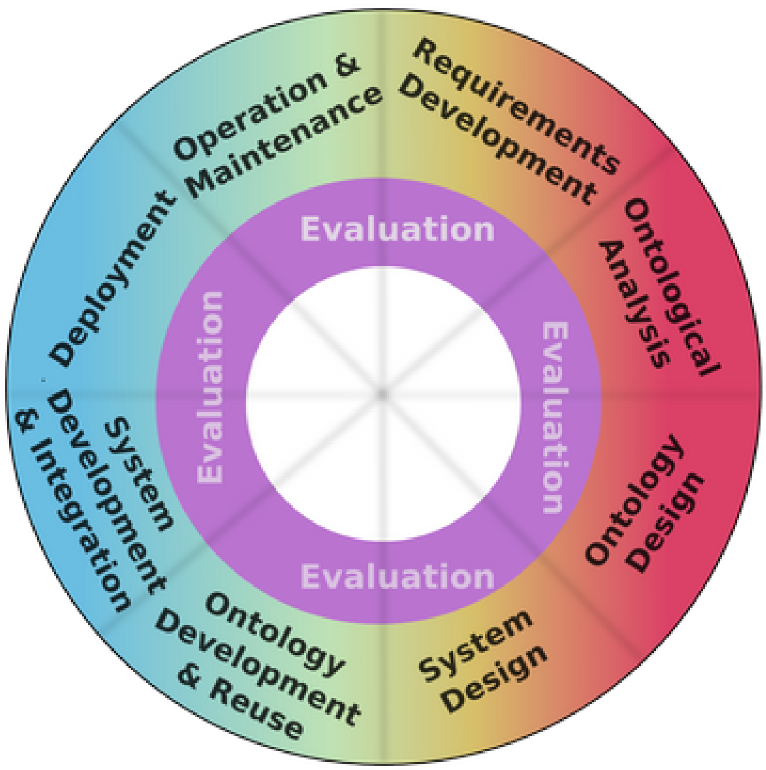

Fig. 1. An Ontology Life Cycle Model

an ontology is not used for machine reasoning as part of a complex piece of software, its evaluation still depends on its intended use within a larger system.

4 Requirements Development Phase

The purpose of this phase is to establish understanding, context, scope, and initial requirements. Development of adequate requirements is critical to the success of any ontology development and usage. Most evaluation activities that are presented in the next sections depend on the results of this phase.

During the *requirements development* phase, expected or intended usages and interpretations are elicited and examined, and initial requirements are derived. Typically, an intended usage is initially understood from a business[4] perspective. The intended usage may be specified as use-cases or scenarios; at early stages, requirements may be captured only as brief statements of one or more business needs and constraints. In many cases only some aspects of usage are addressed, and requirements development may include gathering information about other aspects that are significant for ontology analysis and design.[5]

One important way to specify requirements is by using *competency questions*: questions that the ontology must be able to answer.[6] These questions are formulated in a natural language, often as kinds of queries that the ontology should support in given scenarios.

The output of the requirements development phase is a document that should answer the following questions:

- Why is this ontology needed? (What is the rationale? What are the expected benefits)?
- What is the expected or intended usage (e.g., specified as use-cases, scenarios)?
- Which groups of users needs to understand which parts of the ontology?
- What is the scope of the ontology?
- Are there existing ontologies or standards that need to be reused or adopted?
- What are the competency questions?
- Are the competency questions representative of all expected or intended usages?
- What are the requirements from the operational environment?

[4] "Business" here is meant in the broad sense, incorporating the activities of the organization or user that need the ontology and/or ontology-based system, regardless of whether those activities are commercial, governmental, educational, or other in nature.

[5] See the Ontology Usage reference collection for more about analyzing ontology usage. [4]

[6] See the Competency Questions reference collection for more on capability questions. [5]

- What resources need to be considered during the ontology and system design phases (e.g., legacy databases, test corpora, data models, glossaries, vocabularies, schemas, taxonomies, ontologies, standards, access to domain experts)?

5 Ontological Analysis Phase

The purpose of the *ontological analysis* phase is to identify the key *entities* of the ontology (individuals, classes, and the relationships between them), as well as to link them to the terminology that is used in the domain. This usually involves the resolution of ambiguity and the identification of entities that are denoted by different terms across different resources and communities. This activity requires close cooperation between domain experts and ontologists, because it requires both knowledge about the domain and knowledge about important ontological distinctions and patterns.

The results are usually captured in some informal way, understandable to both ontologists and domain experts. One way of specifying the output of ontological analysis is by a set of sentences in a natural language, which are interpreted in the same way by the involved subject matter experts and ontologists. The ontologists apply their knowledge of important ontological distinctions and relationships to elicit such sentences that capture the information needed to guide the ontology design.[7] Ontological analysis outputs can also be captured in diagrams (e.g., concept maps, UML diagrams, trees, freehand drawings).

The output of the ontological analysis phase, whatever the method of capture, should include specification of:

- significant entities within the scope of the intended usage.
- important characteristics of the entities, including relationships between them, disambiguating characteristics, and properties important to the domain and activities within the scope of the intended usage

[7] An example of such informal outputs is (phrases in italics indicate entities):

Every *pick report* is also an *order status report*.
Every *order* has a *shipping method*.
Possible *shipping methods* include *ground*, and *air*.
The *shipping method* for an individual *order* is determined by the *fulfillment software* after the *order* is *pack*ed.
Every *order* has a *shipping speed*. Possible *shipping speeds* include *standard*, *two-day*, and *overnight*.
The *shipping speed* for a specific *order* is *chosen by* the *buyer* when the *buyer places* the *order*.
For the thing *people in the business* usually call *order*, the *fulfillment database* uses the word "sale."

– the terminology used to denote those entities, and provide enough contextual information to disambiguate polysemous terms.

These results provide input to ontology design and development. In addition, these results provide detail with which high-level requirements for ontology design and development phases can be turned into specific, evaluable requirements.

Evaluating Ontological Analysis Results: Questions to be Answered

The output of an ontological analysis phase should be evaluated according to the following high-level criteria, assisted in detail by the outputs of requirements development:

– Are all relevant terms from the use cases documented?
– Are all entities within the scope of the ontology captured?
– Do the domain experts agree with the ontological analysis?
– Is the documentation sufficiently unambiguous to enable a consistent use of the terminology?

6 Ontology Design Phase

In the *ontology design* phase, a design[8] is developed, based on the outputs from the requirements development and the ontological analysis. In particular, representation languages are chosen for ontology and for queries (these may be identical). Further, the structure of the ontology is determined. Structural choices include whether and how the ontology is separated into modules and how the modules are integrated. As part of the structural design, it may be decided that some existing ontologies are reused as modules. The intended behavior of the modules may be captured by competency questions. These module-specific competency questions are often derived from the ontology-wide competency questions.

Design phase activities include the determination of design principles and of top-level classes in the ontology. The *top-level classes* are the classes in the ontology that are at the highest level of the subsumption hierarchy. (In the case of OWL ontologies, these are the direct children of owl:thing.) These classes determine the basic ontological categories of the ontology. Together the top-level categories and the design principles determine whether and how some fundamental aspects of reality are represented (e.g., change over time). The design

[8] No distinction is made here between design and architecture. The design phase should be understood to encompass both.

principles may also restrict representation choices by the developers (e.g., by enforcing single inheritance for subsumption).

One way to make these design decisions is to use an existing upper ontology. *Upper* or *foundational ontologies* (e.g., DOLCE, BFO, or SUMO) are reusable, varyingly comprehensive ontology artifacts that specify the basic ontological categories, relationships between them, and some methodological decisions about how to represent reality. Other approaches (e.g., OntoClean) rely on the systematic representation of logical and philosophical properties of classes and relationships. There are efforts (e.g., in the NeOn project) to capture design decisions in form of design patterns, and share them with the community.[9]

The results of the design decisions in this phase lead to additional requirements for the ontology. Some of these requirements concern characteristics entirely internal to the ontology itself (e.g., single inheritance for subsumption or distinction between rigid, anti-rigid, and non-rigid classes). Many of these requirements can be understood and evaluated using technical, ontological understanding, without further input of usage-specific or domain-specific information.

Note that there might be conflicting requirements for the expressivity of the ontology language and its performance (see system design phase). Such tension can be addressed by distinguishing between, and developing, separate reference and operational ontologies. A *reference ontology* is one which captures the domain faithfully, to the extent required by the intended or expected usage(s), and in a language that is expressive enough for that purpose. An *operational ontology* is one that is adapted from a reference ontology, potentially incorporating compromises in representation for the sake of performance. The two types of ontologies will be discussed further in the ontology development and reuse section.

Evaluating Ontology Design Results: Questions to be Answered

- Is the chosen ontology language expressive enough to capture the knowledge with sufficient detail in order to meet the ontology requirements?
- Is the chosen query language expressive enough to formalize the competency questions?
- Does the chosen language support all required ontology capabilities? (For example, if the ontology is to support probability reasoning, does the language enable the representation of probabilistic information?)
- Is every individual or class that has been identified in the ontological analysis phase either an instance or a subclass of some top-level class?

[9] See the Existing Methodologies and Upper Ontologies reference collection for some examples of upper ontologies and design methodologies. [6]

- Are naming conventions specified and followed?
- Does the design call for multiple, distinct ontology modules? If so, do the ontology modules together cover the whole scope of the ontology?
- Are all modules of the ontology associated with (informal) competency questions?
- Does the design avoid addition of features or content not relevant to satisfaction of the requirements?
- For each module, is it specified what type of entities are represented in the module (the intended domain of quantification)?
- For each module, is it specified how it will be evaluated and who will be responsible?
- Does the design specify whether and how existing ontologies will be reused?

7 System Design Phase

Information system design as a general activity is its own field of practice, and there is no need to re-invent or summarize it here. There is, however, a need to emphasize the interdependence of ontology design and system design for ontologies that are intended to be used as components of an information system. During system design, decisions are made that lead to requirements for the capabilities and implementation of the ontology and its integration within the larger information system. This interdependency is often underestimated, which leads to poor alignment between the ontology and the larger system it is part of, and thus, to greater risk of failure in ontology and system use.

The output of the system design phase should answer such questions as:

- What operations will be performed, using the ontology, by other system components? What components will perform those operations? How do the business requirements identified in the requirements development phase apply to those specific operations and components?
- What, if any, inputs or changes to the ontology will there be, once the system is deployed?
- What interfaces (between machines or between humans and machines) will enable those inputs? How will these interfaces be tested with respect to the resulting, modified ontology? What requirements will need to be met?
- What data sources will the ontology be used with? How will the ontology be connected to the data sources? What separate interfaces, if any, are needed to enable access to those connections?
- How will the ontology be built, evaluated, and maintained? What tools are needed to enable the development, evaluation, configuration management, and maintenance of the ontology?

- If modularity and/or collaborative development of the ontology are indicated, how will they be supported?

Evaluating System Design Results: Questions to be Answered

The bulk of system design requirements will derive from systems design principles and methodologies in general, and are thus out of the scope of this document. We emphasize here the often unmet need to explicitly recognize the ontology as a component of the system and to evaluate the system design accordingly:

- Does the system design answer the questions listed just above?

8 Ontology Development Phase

The *ontology development* phase consists of four major activities: informal modeling, formalization of competency questions, formal modeling, and operational adaptation (each of which is described below). These activities are typically cycled through repeatedly both for individual modules and for the ontology as whole. In practice, these activities are often performed without obvious transitions between them. Nevertheless, it is important to separate them conceptually, since they have different prerequisites, depend on different types of expertise, and lead to different outputs, which are evaluated in different ways.

The ontology development phase covers both new ontology development and ontology reuse, despite differences between these activities. We do not consider new development and reuse to be part of different phases, for the following reasons: the successful development, or selection and adaptation, of an ontology into an information system is possible only to the extent that the ontology meets the requirements of the expected or intended usage. Thus, whether an ontology is developed entirely from scratch, re-used from existing ontologies, or a combination of the two, good results depend on identification of ontology requirements, an ontological analysis, and the identification of ontology design requirements. Furthermore, the integration of the ontology into the broader information system, its deployment and its usage are not altered in substance by the ontology's status as new or reused. The ontology is evaluated against the same set of requirements, regardless of whether it is reused or newly developed. Therefore, from a high-level perspective, both newly-developed and reused ontologies play the same role within the ontology life cycle.

8.1 Informal Modeling

During *informal modeling*, the result of the ontological analysis is refined. Thus, for each module, the relevant entities (individuals, classes, and their relationships) are identified and the terminology used in the domain is mapped to them. Important characteristics of the entities might be documented (e.g., the transitivity of a relationship, or a subsumption between two classes). The results are usually captured in some informal way (e.g., concept maps, UML diagrams, natural language text).

Evaluating Informal Modeling Results: Questions to be Answered

- All evaluation criteria from the ontological analysis phase apply to informal modeling, with the addition of the following:
- Does the model capture only entities within the specified scope of the ontology?
- Are the defined classes and relationships well-defined? (e.g., no formal definition of a term should use the term to define itself)
- Is the intended interpretation of the undefined individuals, classes, and relationships well-documented?
- Are the individuals, classes, and relationships documented in a way that is easily reviewable by domain experts?

8.2 Formalization of Competency Questions

Based on the results of the informal modeling, the scenarios and competency questions are formalized. This *formalization of competency questions* might involve revising the old competency questions and adding new ones.

Evaluating Formal Competency Questions: Questions to be Answered

- Are the competency questions representative for all intended usages?
- Does the formalization capture the intent of the competency question appropriately?

8.3 Formal Modeling

During *formal modeling*, the content of the informal model is captured in some ontology language (e.g., Common Logic, OWL 2 DL), and then fleshed out with axioms. The resulting reference ontology represents the domain appropriately (fidelity), adheres to the design decisions made in the ontology design phase (craftsmanship), and is supposed to meet the requirements for domain representation (fitness). This is either achieved by creating a new ontology module from scratch or by reusing an existing ontology and, if necessary, adapting it.

Evaluating Formal Modeling Results: Questions to be Answered

The ontology that is developed by the formal modeling activity or is considered for reuse is evaluated in three respects: whether the domain is represented appropriately (fidelity); whether the ontology is well-built and follows the decisions from the ontology design phase (craftsmanship); and whether the representation meets the requirements for its intended use (fitness).

Evaluating Fidelity

Whether the domain is represented accurately in an ontology depends on three questions: Are the annotations of ontology elements (e.g., classes, properties, axioms) that document their intended interpretation for humans (e.g., definitions, explanations, examples, figures) correct? Are all axioms within the ontology true with respect to the intended level of granularity and frame of reference (universe of quantification)? Are the documentation and the axioms in agreement?

Since the evaluation of fidelity depends on some understanding of the domain, it ultimately requires review of the content of the ontology by domain experts.[10] However, there are some automated techniques that support the evaluation of fidelity. For example, one can evaluate the ontology for logical consistency, evaluate automatically generated models of the ontology on whether they meet the intended interpretations,[11] or compare the intrinsic structure of the ontology to other ontologies (or different versions of the same ontology) that are overlapping in scope.

Evaluating Craftsmanship

In any engineering discipline, craftsmanship covers two separate, but related aspects. The first is whether a product is well-built in a way that adheres to established best practices. The second is whether design decisions that were made are followed in the development process. Typically, the design decisions are intended to lead to a well-built product, so the second aspect feeds into the first. Since ontology engineering is a relatively young discipline, there are relatively few examples of universally accepted criteria for a well-built ontology (e.g., syntactic well-formedness, logical consistency and the existence of documentation). Thus, the craftsmanship of an ontology needs to be evaluated largely in light of the ontological commitments, design decisions, and methodological choices that have been embraced within the ontology design phase.

One approach to evaluating craftsmanship relies on an established upper ontology or ontological meta-properties (such as rigidity, identity, unity, etc.),

[10] See the Expert Review and Validation reference collection for more on expert evaluation of ontologies. [7]

[11] See the Evaluating Fidelity reference collection for more on this, including evaluation via simulation. [8]

which are used to gauge the axioms in the ontology. Tools that support the evaluation of craftsmanship often examine the intrinsic structure of an ontology. This kind of evaluation technique draws upon mathematical and logical properties such as logical consistency, graph-theoretic connectivity, model-theoretic interpretation issues, inter-modularity mappings and preservations, etc. Structural metrics include branching factor, density, counts of ontology constructs, averages, and the like.[12]

Evaluating Fitness

The formalized competency questions and scenarios are one source of evidence regarding fitness. These competency questions are used to query corresponding ontology modules and the whole ontology. Successful answers to competency questions provide evidence that the ontology meets the model requirements that derive from query-answering based functionalities of the ontology. The ability to successfully answer competency question queries is not the same as fitness, but, depending on the expected usage, it may be a large component of it.

Fitness can also be evaluated by performing a sample or approximation of system operations, using the ontology in a test environment and/or over a test corpora. For example, if the ontology is required to support automated indexing of documents with ontology terms, then fitness may be evaluated by running an approximation of the document analysis and indexing system, using the ontology in question, over a test corpus. There are various ways of assessing the results, for example, by comparison to a gold standard or by review of results by domain experts, and measured by some suitably defined notions of recall and precision. The extent to which the results are attributable to the ontology, versus other aspects of the system, can be identified to a certain extent by comparison of results using the same indexing system but different ontologies.

8.4 Operational Adaptation

During *operational adaptation*, the reference ontology is adapted to the operational requirements, resulting in an operational ontology. One particular concern is whether the deployed ontology will be able to respond in a time-frame that meets its performance requirements. This may require a paring-down of the ontology and other optimization steps (e.g., restructuring of the ontology to improve performance). For example, it might be necessary to trim an OWL DL ontology to its OWL EL fragment to meet performance requirements.

In some cases the operational ontology uses a different ontology language with a different semantics (e.g., if the application-specific reasoning does not

[12] For more details, see the synthesis and community input pages for intrinsic ontology evaluation [9] and [10].

observe the full first-order logic or description logic Open World Assumption, but instead interprets the negations in the ontology under a Closed World assumption).

Evaluating Operational Adaptation Results: Questions to be Answered

- Does the model support operational requirements (e.g., performance, precision, recall)?

9 System Development and Integration Phase

In this phase the system is built according to the design specified in the design phase. If system components other than the ontology need to be built or otherwise acquired, processes for doing so can occur more or less in parallel to the ontology development phase. Of course, tools and components necessary to the activities in the ontology development phase should be in place as ontology development begins; e.g., ontology development environments, version control systems, collaboration and workflow tools. The system development and integration phase concerns the integration of the ontology and other components into subsystems as called for and into a system as specified in the system design phase.

The system development and integration phase is discussed as part of the ontology life cycle because in a typical application, the functionalities supported by the ontology are realizable not by interaction with the ontology alone, but by processes carried out by some combination of the ontology and other components and/or subsystems. Thus, whether the ontology meets the full range of requirements can only be accurately evaluated once such interaction can be performed and results produced.[13]

Evaluating System Development Results: Questions to be Answered

The bulk of system development requirements will derive from systems development principles and methodologies in general, and are thus out of the scope of this document. We emphasize here the often unmet need to explicitly recognize the ontology as a component of the system and to evaluate the system development results accordingly. Specifically:

- Does the system achieve successful integration of the ontology, as specified in the system design?
- Does the system meet all requirements that specifically relate to the integrated functioning of the ontology within the system?

[13] For more details, see the synthesis page on extrinsic aspects of ontology evaluation. [11]

10 Deployment Phase

In this phase, the ontology goes from the development and integration environment to an operational, live use environment. Deployment usually occurs after some development cycle(s) in which an initial ontology, or a version with some targeted improvement or extension, has been specified, designed, and developed. As described above, the ontology will have undergone evaluation repeatedly and throughout the process to this point. Nevertheless, there may be an additional round of testing once an ontology has passed through development and integration phases and deemed ready for deployment by developers, integrators, and others responsible for those phases. This additional, deployment-phase evaluation may or may not differ in nature from evaluation performed across other life cycle stages; it may be performed by independent parties (i.e., not involved in prior phases), or with more resources, or in a more complete testing environment (one that is as complete a copy or simulation of the operational environment as possible, but still isolated from that operational environment). The focus of such evaluation, however, is on establishing whether the ontology will function properly in the operational environment and will not interrupt or degrade operations in that environment. This deployment-phase testing typically iterates until results indicate that it is safe to deploy the ontology without disrupting business activities. In cases featuring ongoing system usage and iterative ontology development and deployment cycles, this phase is often especially rigorous and protective of existing functionality in the deployed, in-use system. If and when such evaluation criteria have been satisfied, the ontology and/or system version is incorporated into the operational environment, released, and becomes available for live use.

Evaluating Deployment: Questions to be Answered

- Does the ontology meet all requirements addressed and evaluated in the development phases?
- Are sufficient (new) capabilities provided to warrant deployment of the ontology?
- Are there outstanding problems that raise the risk of disruptions if the ontology is deployed?
- Have succeeding competency questions been used to create regression tests?
- Have regression tests been run to identify any existing capabilities that may be degraded if the ontology is deployed? If some regression is expected, is it acceptable in light of the expected benefits of deployment?

11 Operation and Maintenance Phase

This phase focuses on the sustainment of deployed capabilities, rather than the development of new ones. A particular system may have operation and maintenance and new ontology development phases going on at the same time, but these activities should be distinguished as they have different goals (improvement vs sustainment) and they operate on at least different versions of an ontology, if not different ontologies or different modules of an ontology. When an ontology (or version thereof) is in an operation and maintenance phase, information is collected about the results of operational use of the ontology. Problems or sub-optimal results are identified and micro-scale development cycles may be conducted to correct those problems. Simultaneous identification of new use cases, desired improvements, and new requirements that may happen during the same period of use should not be regarded as part of maintenance activity; rather, they are inputs to, or part of, exploration and possibly requirements development for a future version, extension, new ontology or new module. A single set of tools may be used to collect information of both sorts (for maintenance and for forward-looking exploration and requirements development) while an ontology is in use, but the information belongs to different activities. This distinction is manifested, for example, in the distinction between "bug reports" (or "problem reports") and "feature requests" (or "requested improvements") made by bug-tracking tools. The maintenance activity consists of identifying and addressing bugs or problems.

Evaluating Operation and Maintenance: Questions to be Answered

The evaluation should be continuous, e.g., open problem reporting and regular, e.g., nightly, automated regression testing:

- Are any regression tests failing? If so, are they being addressed?
- Is any functionality claimed for the most recent deployment failing? If so, can the problem be tracked to the ontology, or is the problem elsewhere?
- If the problem is located in the ontology, can it be corrected before the next major development and deployment cycle? If so, is it being addressed?
- If a problem occurs and cannot be addressed without a large development cycle effort, is the problem severe enough to warrant backing out of the deployment in which it was introduced?

12 Tools for Ontology Evaluation

There are central aspects of ontology that may not be amenable to software control or assessment. For example, the need for clear, complete, and consistent lexical definitions of ontology terms is not presently subject to effective software consideration beyond identifying where lexical definitions may be missing entirely. Another area of quality difficult for software determination is the fidelity of an ontology.

There are no tools for ontology development or to enable ontology evaluation across the whole life cycle. Existing tools support different life cycle phases, and for any given characteristic, some tools may perform better in one phase than in another phase where a different tool is better suited. However, significant new ontology evaluation tools are currently becoming available to users.[14] An overview is presented as part of the Ontology Quality Software Survey.[15]

13 Observations and Recommendations

1. We still have a limited understanding of the ontology life cycle, ontology development methodologies, and how to make best use of evaluation practices. More research in these areas is needed. Thus, any recommendation in this area is provisional.

2. There is no single ontology life cycle with a fixed sequences of phases. However, there are recurring patterns of activities, with identifiable outcomes, which feed into each other. In order to ensure quality, these outcomes need to be evaluated. Thus, evaluation is not a singular event, but should happen across the whole life of an ontology.

3. The different outputs of the the ontology life cycle phases need to be evaluated with respect to the appropriate criteria. In particular, different requirements apply to informal models, reference ontologies, and operational ontologies, even when implemented in the same language.

4. Ontologies are evaluated against requirements that derive both from design decisions and the intended use of the ontology. Thus, a comprehensive evaluation of an ontology needs to consider the system that the ontology is part of.

5. There is a shortage of tools that support the tracking of requirements for and the evaluation of ontologies across all stages of their development and use. These kinds of tools should be developed, and integrated in ontology development environments and repositories.

6. We strongly encourage ontology developers to integrate existing evaluation methodologies and tools as part of the development process.

[14] See the Tool Support reference collection [12] and the Ontology Summit synthesis page on software environments [13] for more information about available tools.

[15] For example. For Survey and results, see the Software Support for Ontology Quality and Fitness Survey.

This communiqué was endorsed by the following 162 members of the ontology community.[16]

Sameera Abar
Tara Athan
Nathalie Aussenac-Gilles
Ken Baclawski
Mary Balboni
Michael Barnett
John Bateman
Denise Bedford
Joel Bender
Mike Bennett
Gary Berg-Cross
Julita Bermejo-Alonso
Olivier Bodenreider
Harold Boley
Nikolay Borgest
Stefano Borgo
Michel vanden Bossche
Bruce Bray
Mathias Brochhausen
Rex Brooks
Peter Brown
Pat Cassidy
Francisco E. Castillo-Barrera
Vinay Chaudhri
Chris Chute
Anthony Cohn
Mills Davis
Mike Dean
Michael Denny
Nicolau DePaula
Jim Disbrow
Ed Dodds
Michel Dumontier
Astrid Duque Ramos
Karen Engelbart
Doug Engelbart
Jesualdo T. Fernández Breis
Tim Finin
Michael Fitzmaurice
Elizabeth Florescu
Doug Foxvog
Gilberto Fragoso
Bart Gajderowicz
Aldo Gangemi
Gary Gannon
Katherine Goodier
Henson Graves
Tom Gruber
Michael Gruninger
Giancarlo Guizzardi
Melissa Haendel
Torsten Hahmann
Marc Halpern
Nurhamizah Hamka
Ali Hashemi
Brian Haugh
Martin Hepp
Matthew Hettinger
Scott Hills
Ralph Hodgson
Bill Hogan
Jeanne Holm
Doug Holmes
Ian Horrocks
Dosam Hwang
Kingsley Idehen
Megan Katsumi
Maria Keet
Elisa Kendall
Pavithra Kenjige
Kyoungsook Kim
Mitch Kokar
Pradeep Kumar
Oliver Kutz
Christoph Lange
Ken Laskey
David Leal
Kiyong Lee
Anatoly Levenchuk
Antonio Lieto
Laurent Liscia
Frank Loebe
Terry Longstreth
Joanne Luciano
Ernie Lucier
Mark Luker
Dickson Lukose
Deborah MacPherson
Diego Magro
Patrick Maroney
Richard Martin
Howard Mason
Bill McCarthy
John McClure
Tim McGrath
Chris Menzel
Riichiro Mizoguchi
Till Mossakowski
Enrico Motta
Regina Motz
Mark Musen
John Mylopoulos
Joel Natividad
Fabian Neuhaus
David Newman
Deborah Nichols
Duane Nickull
Pete Nielsen
Leo Obrst
Frank Olken
Alessandro Oltramari
Mary Parmelee
Chris Partridge
Vinod Pavangat
Lynne Plettenberg
Hans Polzer
María Poveda Villalón
David Price
Fouad Ramia
Steve Ray
Alan Rector
Michael Riben
Jack Ring
Carlos Rueda
Arturo Sanchez
Bob Schloss
Todd Schneider
James Schoening
Ravi Sharma
Bradley Shoebottom
Barry Smith
Bob Smith
Jerry Smith
Dagobert Soergel
Richard Mark Soley
John Sowa
Simon Spero
Jim Spohrer
Christopher Spottiswoode
Ram Sriram
Mark Starnes
George Strawn
Rudi Studer
Samir Tartir
Hans Teijgeler
Andreas Tolk
Meika Ungricht
Michael Uschold
Marcela Vegetti
Laure Vieu
Amanda Vizedom
Evan Wallace
James Warren

[16] Please note that these people made their endorsements as individuals and not as representatives of the organizations they are affiliated with.

Chris Welty	Nancy Wiegand	Peter Yim
Matthew West	Tim Wilson	Cecilia Zanni-Merk
Trish Whetzel	Dennis Wisnosky	Marcia Zeng

Links

[1] http://ontolog.cim3.net/cgi-bin/wiki.pl?OQuaRE
[2] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/PMKFZPDA
[3] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/HU2MCEG4
[4] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/HJ6MK7W3
[5] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/7B5TCZCZ
[6] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/FVM3J9FJ
[7] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/6GGPKU3D
[8] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/929KF23Z
[9] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Intrinsic_Aspects_Of_Ontology_Evaluation_Synthesis
[10] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Intrinsic_Aspects_Of_Ontology_Evaluation_CommunityInput
[11] http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Extrinsic_Aspects_Of_Ontology_Evaluation_Synthesis
[12] https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/DWNMSJ5S
[13] OntologySummit2013_Software_Environments_For_Evaluating_Ontologies_Synthesis

Section 3
Going Forward

The Role of Ontologies for Evaluation of Standards

Steve Ray

The 2013 Ontology Summit was dedicated to the evaluation of *ontologies*. Centrally related to the mission of NIST, we can build on this examination to consider how ontologies can help in the specification and evaluation of *standards*. Normative information standards strive to be unambiguous specifications of terms, their definitions and relationships, and their intended use. This is of course more or less synonymous with the definition of an ontology, such as Tom Gruber's definition.One is therefore led to the suggestion that specifying a standard in the form of an ontology might be the best way to capture those intended specifications, and in this way the evaluation of the ontology is also the evaluation of the standard. Representing a standard as an ontology has indeed been tried, with perhaps the most formal example being the normative specification for ISO 18629, known as the Process Specification Language, or PSL. This example is illustrative in that the specification provably self-consistent (not an easy task with other normative standards).

What the PSL example shows is that once a standard is represented in ontological form, certain properties such as self-consistency become easy to validate. To tap into the power of ontology to evaluate standards, one can either define the standard natively as an ontology, or one can convert a standard from some other representation into an ontology. This latter tactic has also been tried and shows great promise for improving the quality of many standards currently represented in languages such as XSD, UML, or EXPRESS.

1 Next Steps for NIST

To more effectively fulfill its mission in providing the USA with a set of high-quality information standards, NIST could pursue several strategies:

1. Build an environment and toolkit for the evaluation of standards, to include:
   - Transformation tools to convert existing standards in various representations into ontologies, retaining provenance information for backward referencing the original standard
   - Testing tools to evaluate the converted standards, (building directly on the results of the 2013 Ontology Summit), including reporting and recommendation tools
2. Build an environment and toolkit for the authoring of standards as native ontologies:

- Inclusion of naming and design rule tools to assist in good design practice
- Support for multiple domain-specific views of the model structure (similar to capabilities offered in commercial UML modeling tools today supporting model-based software engineering).

The two strategies listed above represent the creation of user-facing toolkits that would not require the user to be an expert in ontological representations. This is in contrast to the examples listed above, where both the authoring and the conversion activities required significant expertise in the use of ontology languages and concepts. PSL, for example, was written primarily by an expert in first-order logic, using little more than a simple text editor.

Towards an Ontology Evaluation Testbed

Fabian Neuhaus

1 Introduction

Ontologies allow the reusable, human understandable, and machine interpretable representation of knowledge. They are used in for a broad range of purposes, including system integration, data mining, natural language processing, and standardization of terminologies.[1] This versatility of ontologies is partially the result of the ability to represent knowledge in an accessible way without loosing its complexity. However, this means that the complexity of the domain that is represented is mirrored by the ontology. E.g., the Foundational Model of Anatomy (FMA), which is an ontology of the canonical human anatomy, consists of more than 83 000 different classes which are connected by 183 different relationship types, and more than 2.5 million relationship instances [2,16]. Given the size and internal complexity of many ontologies (like the FMA) maintaing quality is a difficult challenge.

During the Ontology Summit 2013 the ontology community took stock of the situation of ontology evaluation (see page 2). This position paper is a response to some of the findings of the Ontology Summit. It outlines a path forward, which would both (i) enable a better use of existing metrics and tools for ontology evaluation, and (ii) support the development of new measurement science for ontology evaluation. The goal is to improve the process of ensuring ontologies behave correctly, to identify defects earlier, and to help industry to reduce the cost for ontology development and maintenance.

2 The Puzzling State of Ontology Evaluation

Concerning the current state of ontology evaluation, one can make three observations:

There is no lack of research on ontology evaluation. This is easily verified by a quick search on the internet: a popular search engine finds 2430 scientific articles with the keyword "Ontology Evaluation" [1]. The literature discusses many different techniques and approaches on how to evaluate ontologies and improve their quality.

[1] A list of examples for use cases of ontologies can be found here [3].

Ontologists believe that quality of ontology is important. One way to support this claim is by studying the evolution of ontologies [8]. Many changes that are made are motivated by the goal to remove errors or improve the architecture of the ontology. If developers of ontologies would not care about the quality of their ontologies, they would not make these changes.

Ontology evaluation plays a minor role in ontology development. This is supported by the fact that publications about ontologies often lack any evaluation of the ontology. Our personal observation is that in many cases ontology developers utilize consistency checks during the development process, and otherwise rely on feedback from their users to improve the quality of their ontologies.

Together, these three observations are rather puzzling. Ontologists seem to ignore evaluation methodologies and techniques that would enable them to achieve what they want, namely building quality ontologies. The question is why they are not adopting the available ontology evaluation techniques. After interviewing members of the community, we identified the following four challenges:

Challenge 1. Lack of agreement on evaluation and quality. There has been no community consensus on development methodology, the notion of 'quality' for ontologies, let alone on suitable metrics, or measurement techniques. Consequently, while there are many papers on individual approaches, there is no shared 'bigger picture' on the aspects of ontologies that should be evaluated, how the techniques related to each other, and when the evaluations should happen during the ontology development process. This lack of agreement discourages the adoption of evaluation techniques.

Challenge 2. Lack of evaluation of proposed metrics. There is a wide range of proposed ontology evaluation methods and metrics, but at this time there there is little work on the evaluation of the proposed metrics themselves. (This is in particular problematic, because some of the proposed metrics seem to have a rather dubious value for measuring quality; e.g., the depth of the subsumption hierarchy of an ontology.) The situation is made more difficult by the fact that the importance of a given metric depends often on the intended use case for the ontology and, thus, the external requirements for the ontology.

Challenge 3. Lack of tools. Given the number of publications on ontology evaluation, there is a surprising lack of available tools. A survey during the Ontology Summit identified 15 tools based on a very wide notion of 'ontology evaluation' [4].

Challenge 4. Barriers for use. Many of the existing tools are stand-alone tools that are the result of research projects, and people are not aware of them. And since they are not integrated in tools that are already in use, there is a barrier to integrate these evaluation tools into existing workflows.

Challenge 5. No support of acceptance testing. Ontologies are build as part of information systems.[2] Both from a system engineering and quality management perspective the most important kind of ontology evaluation would be *acceptance testing*, which evaluates whether the ontology meets the specified requirements, and, thus, is able to perform as intended in the specified information system. – Unfortunately, most approaches for ontology evaluation rely on either logical properties of the ontology (e.g, consistency) or structural properties of the ontology (e.g., depth of subsumption hierarchy) or metadata (e.g., comments by users). The structural and logical properties are sometimes evaluated with the help of additional resources, like other ontologies or text-corpora or data sets. However, usually the evaluation process does not take into account the use-case and the intended function of a given ontology. Two notable exceptions are OntologyTest [12] and the XD selector [5], which support the use of formalized competency questions as a means to evaluate the functional behavior of an ontology [11,13]. However, even these tools do not link functional test to the requirements for the ontology. Therefore, the existing tools do not support the kind of evaluation that would be most useful from a system engineering and quality management perspective.

3 Towards an Ontology Development Testbed

The Communique of the Ontology Summit 2013 establishes a framework for the different aspects of quality of ontologies across the whole life cycle of an ontology, which has been endorsed by more than 160 ontologists. This will, hopefully, contribute to overcome the Challenge 1 in the sense that it establishes a shared conceptualization, which allows to locate the role of a proposed ontology evaluation technique within the ontology life cycle.

To overcome Challenges 2-5 for ontology evaluation, we propose the development of an ontology testbed, which consists of three major components: a repository, a tool library containing both evaluation engines and tools that are common between evaluation engines, as well as an ontology test manager (see Figure 1).[3] One immediate benefit of having one testbed that supports a broad

[2] 'Information system' is used here in a board sense, which may include humans.

[3] A first step towards such a testbed has been made by the integration of OOPS! and Ontohup (see page 47, further [14,15]).

variety of evaluation tools is that it allows to easily compare the results of the evaluation approaches. This will enable a better comparison of the merits of the proposed metrics, and, thus address Challenge 2.

3.1 Repository

One key insight of the Ontology Summit has been that any evaluation of an ontology depends, at least partially, on the intended use of the ontology as part of an information system. Thus, for a meaningful evaluation of an ontology one needs to link the ontology to its requirements. If one takes this system view on ontology evaluation seriously, it means that ontology evaluation is not just concerned with ontologies, but with a whole trace of intermediary documents that are generated during the ontology development process. These include *behavior specifications* of the ontology, which are used to capture requirements and design decisions. An important way to specify behavior of ontologies is the combination of *scenarios* and *competency questions*. Further, a *terminology*, which covers the terms from the scenarios that need to be captured in the ontology. Another document may be an *informal model* as an intermediate step to the formal ontology. All these documents are not formal ontologies, but supposed to be written in a way that is accessible to domain experts. These documents are the basis for building a *reference ontology*, which, if necessary, may be adapted to an *operational ontology*. Last, but not least, assuming the behavior specification involved competency questions, there are 'emphformalization of the competency questions, which can be used to validate whether the axioms in an ontology are strong enough to support the intended behavior.

The repository should be extension of an Open Ontology Repository (OOR) [6], which contains not only ontologies but the other kinds of documents listed above. The representation of the repository in Figure 1 is somewhat simplified, since it leaves many functionalities of an OOR implicit. E.g., an OOR provides version control for ontology, supports ontology modularization and mapping, allows the representation of the same ontology in different languages, and contains metadata of the ontologies.

This information provides a much richer foundation for the evaluation of an ontology than a single snapshot of the ontology would provide. In particular, keeping track of the requirements for an ontology enables the support of acceptance testing (Challenge 5). Further, assuming that the ontology testbed is build on top of an OOR that is already in use, the barrier for adopting ontology evaluation tools is lowered (Challenge 3), since from the perspective of ontology developers the tools will be just extending the functionality of a the repository that they already use.

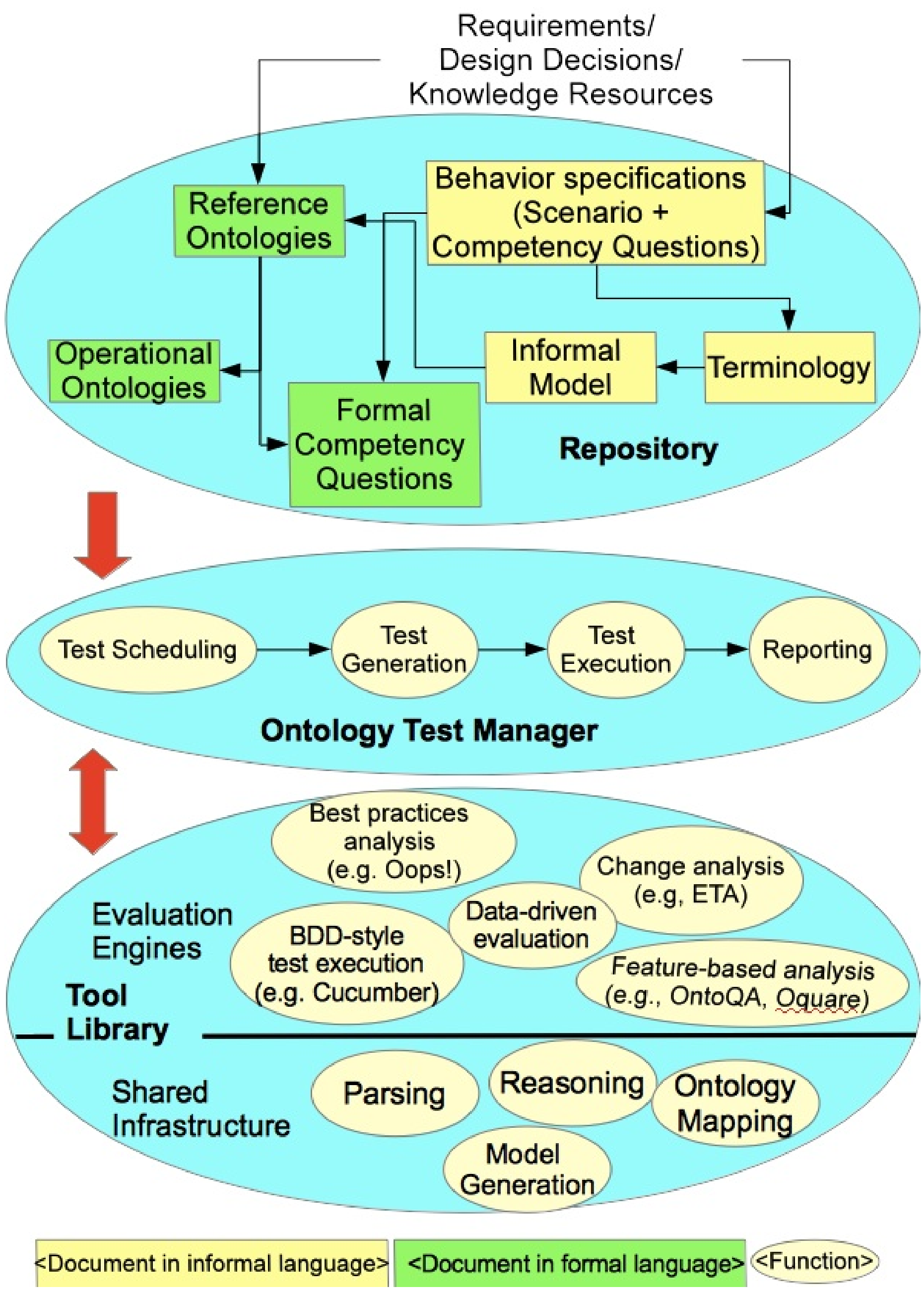

Fig. 1. Functional Architecture of the Ontology Testbed

3.2 Tool Library

As we observed, although there is a lot of literature on ontology evaluation, there is a surprising lack of tools for ontology evaluation available (Challenge 3). One goal of the testbed is to make the development of new tools easier by allowing developers to focus on the evaluation engines, while reusing the ontology test manager and tools for shared functionalities; for example, parsing, automatic reasoning, model generation, or ontology mapping.

As mentioned above, a wide variety of approaches for ontology evaluation are discussed in the literature, and should be supported by the testbed. In Figure 1 we try to illustrate the point by providing a few examples: analyzing an ontology based on a collection of best practices (e.g., with (Oops! [15]), data-driven evaluation based on a text corpus (as discussed in [7]), analyzing change between versions of the same ontology (e.g., Evolutionary Terminology Auditing (ETA) [8]),and an analysis of the structural features of the ontology (e.g., with OntoQA [17,18], Oquare [9,10]).

In the Ontology Summit Communique scenarios and competency questions are mentioned specifically as a means to capture functional requirements for the content of an ontology. As mentioned above, we consider these as a specific cases of behavior specifications, as used in Behavior Driven Development (BDD) in software engineering. These behavior specifications may not only be concerned with the content of ontologies (as competency questions are), but, for example, the performance of ontologies (in combination with a given automatic reasoning engine). To execute these kinds of BDD-style tests, the tool library needs a BDD-test execution engine, like Cucumber [19].

3.3 Ontology Test Manager

This component interacts with both the repository and the tool library for test scheduling, text generation, test execution, and reporting. In the case of complex modular ontologies, the component for test scheduling needs to keep track between dependencies between ontologies.

References

1. http://scholar.google.de/scholar?hl=en&q=%22ontology+evaluation%22+&btnG=&as_sdt=1%2C5&as_sdtp=
2. Foundational Model of Anatomy, http://bioportal.bioontology.org/ontologies/FMA
3. Ontology Summit 2011 CaseStudy Use Case Matrix, http://ontolog.cim3.net/file/work/OntologySummit2011/ValueMetrics/OntologySummit2011-CaseStudy--UseCaseMatrix.pdf

4. Ontology Summit2013 Survey, http://ontolog-02.cim3.net/wiki/Category:OntologySummit2013_Survey
5. XD Tools documentation, http://stlab.istc.cnr.it/documents/XD/documentation/1.1/
6. Baclawski, K., Schneider, T.: The open ontology repository initiative: Requirements and research challenges. In: Proceedings of Workshop on Collaborative Construction, Management and Linking of Structured Knowledge at the ISWC (2009)
7. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data driven ontology evaluation. In: International Conference on Language Resources and Evaluation (LREC 2004) (2004)
8. Ceusters, W.: Applying evolutionary terminology auditing to the Gene Ontology. Journal of biomedical informatics 42(3), 518–529 (2009)
9. Duque-Ramos, A., FernáNdez-Breis, J.T., Iniesta, M., Dumontier, M., EgañA Aranguren, M., Schulz, S., Aussenac-Gilles, N., Stevens, R.: Evaluation of the OQuaRE framework for ontology quality. Expert Syst. Appl. 40(7), 2696–2703 (Jun 2013), http://dx.doi.org/10.1016/j.eswa.2012.11.004
10. Duque-Ramos, A., Fernández-Breis, J.T., Stevens, R., Aussenac-Gilles, N.: OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. Journal of Research and Practice in Information Technology 43(2), 159 (2011)
11. Fox, M.S., Gruninger, M.: Enterprise modeling. AI magazine 19(3), 109 (1998)
12. García-Ramos, S., Otero, A., Fernández-López, M.: OntologyTest: A Tool to Evaluate Ontologies through Tests Defined by the User. In: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, pp. 91–98. Springer (2009)
13. Grüninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Benchmarking Theory and Practice, pp. 22–31. Springer (1995)
14. Mossakowski, T., Kutz, O., Codescu, M.: Ontohub-a repository engine for heterogeneous ontologies and alignments, http://ontolog.cim3.net/forum/ontoiop-forum/2013-05/pdfqyVdCx7UaU.pdf
15. Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: Validating ontologies with OOPS! In: Knowledge Engineering and Knowledge Management, pp. 267–281. Springer (2012)
16. Rosse, C., Mejino Jr, J.L.: A reference ontology for biomedical informatics: the Foundational Model of Anatomy. Journal of biomedical informatics 36(6), 478–500 (2003)
17. Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P., Aleman-Meza, B.: OntoQA: Metric-based ontology quality analysis. In: IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources. vol. 9 (2005), http://knoesis.org/library/download/OntoQA.pdf
18. Tartir, S., Budak Arpinar, I.: Ontology evaluation and ranking using OntoQA. In: Semantic Computing, 2007. ICSC 2007. International Conference on. pp. 185–192. IEEE (2007)
19. Wynne, M., Hellesoy, A.: The cucumber book: behaviour-driven development for testers and developers. Pragmatic Bookshelf (2012)