VIEWERS PLEASE NOTE: 2nd draft, for review during 2013-04-18 Summit Session, is at:

https://docs.google.com/document/d/1jeU5DL9JuNiyxv7BDOMr0VgFlWmSKVHXcnLlcHF-Zg0/edit?usp=sharing

Ontology Quality and Its Evaluation

Executive Summary

[To be added later. The intended audience of the executive summary is wider than the intended audience for the rest of the text.]

[Add: purpose of document]

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 considered as a type of glossary. Since ontologies capture key concepts and their relationships in a machine interpretable form, they are similar to domain models of system 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 characteristics of an ontology, comparison of the results with a set of ontology requirements, and assessment of the ontology's suitability for some purpose. Some ontology characteristics can be measured independent of usage; others involve how an ontology relates to its intended domain, environment, or usage-specific activity, and thus can only be measured with reference to some usage context. Moreover, measurement alone does not make an evaluation. Evaluation of an ontology requires: identifying which possible ontology characteristics are relevant to the intended usage and what requirements must therefore be met by the ontology; measuring these characteristics; and determining, on this basis, to what degree the ontology fits the requirements for that intended usage. The variety of the potential uses of ontologies means that there is no single list of requirements and no single approach in evaluating ontologies against them.

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

four high-level characteristics:

(1) Can humans understand the ontology correctly? (Intelligibility)
(2) Does the ontology represent the domain appropriately? (Model fidelity)
(3) Does the representation of the domain fit the requirements for its intended use? (Model fitness)
(4) Does the deployed ontology meet the requirements of the information system of which it is part? (System fitness)

For intelligibility it is not sufficient that ontologists can understand the content of the ontology; the definitions and axioms need to be transparent to all intended users. This may require multiple annotations of an element of the ontology suitable to for different audiences. Model fidelity encompasses a wide range of aspects, including logical consistency, structural soundness of the ontology, and factual correctness. Both model fitness and system fitness are dependent on requirements for the intended usage; these requirements might derive from the operational environment in the case of an ontology that is deployed as part of an information system or the goals for the knowledge representation project if the ontology is deployed as a standalone reference ontology.

The purpose of this document is to provide an overview of the activities that need to occur during the phases of a life cycle of an ontology, and the critical relationship between life cycle phase activities, evaluation, and the quality of the result. In the next section we present a breakdown of the ontology development life cycle. In the following sections we identify some of the activities that occur during each phase, the characteristics of the ontology that should be evaluated at the stage, and the applicable ontology evaluation methods. The document concludes with some observations about the current tool support for ontology evaluation, and recommendations for future work.

An Ontology Life Cycle Model

The life cycle of an ontology is the succession of phases in which the ontology is being circumscribed, specified, developed, deployed, and used. While there is no single sequence that all ontologies follow, there are identifiable phases through which ontologies pass, usually iteratively. There are also phases, sequences, and iterations through which an ontology should go, for best quality and results in use.

The figure below presents a schematic view of the ontology life cycle. While dependencies exist between some pairs of stages, there is not a single, necessary sequence of steps that characterizes the ontology life cycle. Moreover, ontology life cycles may vary, as software life cycles do, according to particular project methodologies that incorporate specific sequences, iterations or other process-organizing features. These sequences may themselves be overlapping; for example, exploration of a later release may begin while an earlier release is being built. Information may then flow not only in more than one direction across phases, but across entire sequences and cycles.

include better figure depicting the ontology lifecycle

VOLUNTEERS WANTED
Life cycle phases
- requirements development and analysis
- ontological analysis

- ontology design
- system design
- ontology development & reuse
- system development & integration
deployment
- operation and maintenance

figure 1

This presentation of the ontology life cycle assumes a situation, where an ontology is developed for the use of some information system. Hence, it needs to be integrated into the rest of the system and may need to be adapted to performance requirements of that system. However, these steps are not necessary during the development of a reference ontology that is developed to represent a domain without any specific software application in mind. For these kind of reference ontologies the system design, system development and integration, and deployment phases are omitted.

In the life cycle model we do not distinguish between ontology development and reuse as separate phases. One reason is that it is rarely possible to reuse a given ontology without adapting it or integrating it into a larger ontology. More importantly, the successful reuse of an existing ontology within an information system presupposes that the ontology meets the requirements of the intended usage. Thus, the evaluation of the suitability of an ontology for reuse does not differ significantly from the evaluation of new ontology during its development process: both need to meet the requirements identified during the requirements development and analysis, ontological analysis, ontology design, and system design phases.

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 these phases is discussed in more detail and linked to the phase-appropriate evaluation activities. The evaluation provides actionable information regarding the degree to requirements of the life cycle phase are being met. Requirements identification will be discussed in greater detail in the next section.

Requirements Development and Analysis

The purpose of this phase is to establish understanding, context, scope, and initial requirements. During this phase, the intended usage is examined, and requirements are derived from that usage. Typically, an intended usage is initially understood from a business[1] 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. The requirements development and analysis phase involves extending and clarifying initial information until the intended usage is sufficiently captured and understood to effectively guide technical decisions. This process involves an interplay of technical, business, and project-sponsor understanding. Adequate requirements development and analysis is critical to the success of any ontology development or usage. Because it is here that the grounding requirements for the ontology are identified, against which candidate ontologies can be meaningfully evaluated.

Requirements Development and Analysis begins with a focus on understanding what

[1] “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.

requirements Some ontology requirements will concern the relationship of the ontology and aspects of the operational environment. Such requirements might involve, for example: adequate scope of ontology coverage; inclusion of ontology content or logical features supporting particular systems operations; relationships to applicable standards and to other ontologies; and intelligibility of the ontology to intended users.

Requirements development and analysis will also identify requirements of a different kind: those that are entirely independent of the content or internal characteristics of the ontology. These characteristics, (such as access and security thereof, licensing conditions, cost) can be evaluated without technical ontological understanding. For this reason, the remainder of this document focuses on evaluation of the first two types, in which specifically ontological considerations apply.

The output of the requirements development and analysis phase is a document that answers the following questions:
- Why do we need an ontology?
- What is the intended usage (e.g., specified as use-cases, scenarios)?
- What is the scope of the ontology?
- What are the requirements for domain representation (requirements for model fitness)?
- What are the competency questions?
- What are the requirements from the operational environment (requirements for system fitness)?
- What are resources that 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)?

Ontological Analysis

The purpose of this 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 involves usually the resolution of ambiguity and the identification of entities that denoted by different terminology within different resources and communities.

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 very consistently 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).[2] -- Ontological analysis outputs can also be captured in diagrams (e.g., concept maps, UML diagrams, trees, freehand drawings).

[2] An example of such informal outputs is:
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 the people in the business usually call *order* the fulfillment database uses the word ‘sale’.

Whatever the method of capture, the results of ontological analysis should specify:

- the significant entities within the scope of the intended usage.
- the important properties, relationships, and disambiguating characteristics of these entities.
- where entities can best be distinguished by noting their relationship to something outside of the domain, specification of such relationships and the domain of the out-of-scope entities references (e.g., to distinguish two entities associated with the expression “report,” it might be noted that one is a type of (a plain-English or common-sense notion of) activity, while the other is a type of (a plain-English or common-sense notion of) object or artifact or piece of documented information.
- the terminology used to denote those entities.

Evaluation

- 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?

Ontology Design Phase

In this phase, the requirements from the requirements analysis phase are used to guide basic modeling design decisions. Such choices include the ontology language to be used, the structure of the ontology, any upper ontology to be used and ontology design principles. Choices of upper ontology and design principles determines such things as: whether and how the ontology incorporates high level ontological categories; whether and how some fundamental aspects of reality (e.g., change over time) are represented; and which methodological practices will hold (e.g., single or multiple inheritance for subsumption). The structure of the the ontology includes such things as whether and how the ontology is separated into modules, and how such modules interact or are integrated.

The modelling design decision are dependent on the business requirements identified in the requirement phase, but also add additional requirements that the ontology needs to meet. Some of these requirements concern characteristics entirely internal to the ontology itself (e.g., syntactic well-formedness, logical consistency, modularity, single inheritance for subsumption). These requirements can be understood and evaluated using technical, ontological understanding, without further input of usage-specific or domain-specific information.

The competency questions that are formulated in the exploration phase capture in natural language the kinds of queries that the ontology should support in given scenarios. The role of the individual ontology modules is clarified by using the ontology-wide competency questions to formulate for each ontology module scenarios and competency questions that capture the intended behavior of the module.
Note that there might be conflicting requirements for the expressivity of the ontology language and its performance (see system design phase). In this case one needs to distinguish between the reference ontology, which represents the domain faithfully in a language that is expressive enough for that purpose, and the implementation ontology, which might compromise the representation of the domain for the sake of performance.

Evaluation

- Is the chosen ontology language expressive enough to capture the knowledge sufficiently detailed to meet the requirements identified in the Ontology Analysis phase?
- Do the upper ontology and design principles meet established best practices?
- Do the ontology modules together cover the whole scope of the ontology?
- Are the competency questions representative for all intended usages?
- Are all modules of the ontology associated with competency questions?

System Design Phase

The system design phase involves basic design decisions about how the ontology is implemented and integrated within the larger information system. This phase can occur in parallel with the ontology design phase. The intended usage of the ontology (through all life cycle stages), the means by which users interact with it, the operations to be performed using it, and the outputs of these operations all entail requirements not only the for ontology-as-domain-model but for the ontology-as-software-artifact, and the larger system(s) within which the ontology is incorporated. The output of the system design phase should answer such questions as:

- 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 ontology correctness? What requirements will need to be met?
- What, if any, data sources, will be 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, and maintenance of the ontology?
- If modularity and/or collaborative development of the ontology are indicated, how will they be supported?
- 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 and analysis phase apply to those specific operations and components?

Evaluation

While the design of ontology-enabled systems has some specific steps or considerations, noted above, the evaluation of resulting systems design(s) should follow best practices for evaluation of information and software system design in general.

Ontology Development and Reuse Phase

The ontology development phase consists of four major activities: Informal Modeling, Formalizing Competency Questions, Formal Modeling, and Operational Adaption. 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.

While this section is written with the main focus on ontology development, the evaluation of a candidate ontology for reuse follows the same processes, and is based on the same requirements. The difference is that the evaluation results during development provide feedback to the development team, while in the evaluation of ontologies for reuse provides information about the degree to which the ontology in question meets the requirements of the intended usage.

Informal Modelling

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).

Evaluation

- All evaluation criteria from the ontological analysis phase.
- Are no entities outside the scope of the ontology captured?
- 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?

Formalizing Competency Questions

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

Evaluation

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

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 and is supposed to meet the requirements for domain representation (model fitness). This is either achieved by creating a new ontology module from scratch or by reusing an existing ontology and, if necessary, adapting it. The reference ontology is more likely to be reusable than the implementation ontology that is the result of operational adaptation (see below).

Evaluation

The ontology that is developed by the formal modelling activity is evaluated in two respects: Is the domain represented appropriately (model fidelity) and does the representation meet the requirements for its intended use (model fitness).

Evaluating Model Fidelity

Model fidelity is determined by two questions: Does the ontology follow best practices; in particular does it implement the upper ontology and the design principles decisions made in the model design phase? And is the domain represented accurately, that is are all axioms within the ontology true?

Model fidelity is often evaluated by examining the intrinsic structure of one ontology or comparing the intrinsic structure of several ontologies that are overlapping in scope. These kinds of evaluation techniques draw 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.[3]

Another set of techniques for the evaluation of model fidelity involves some understanding of the domain. This is often required to determine whether a particular axiom is in alignment with the reality it is supposed to model, whether the model captures the distinctions and properties important to the domain, or which ontological design principles to apply in a given situation. Some ontological meta-properties (such as rigidity, identity, unity, etc.) can be used to gauge the quality of the axioms of the ontology.

Evaluating Model Fitness

The formalized competency questions and scenarios are used to query the ontology modules and the whole ontology. The queries will only be successful if the axioms are sufficiently strong enough to rule out unintended interpretations satisfying the ontology. If all competency questions are answered successfully, the ontology is complete with respect to the competency questions. This is evidence that the ontology meets the requirements for domain representation that derive from functionalities of the ontology that rely on query-answering.

As appropriate to the requirements identified during requirements identification and development and ontology analysis and ontology and system design phases, model fitness may be also be evaluated in part 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 review of results by domain experts. 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.

Operational Adaptation

The implementation ontology is the result of adapting the reference ontology to the operational requirements. One particular concern is whether the deployed ontology will be able to respond in a time-frame that meets its performance requirements (system fitness). This often requires 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 implementation ontology uses a different ontology language with a different

[3] For more details, see: http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Intrinsic_Aspects_Of_Ontology_Evaluation_Synthesis

semantics. E.g., if the application-specific reasoning does not observe the full first-order logic or description logic Open World Assumption, but instead the negations in the ontology under a Closed World assumption.

Evaluation

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

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 the full range of requirements can only be accurately evaluated once such interaction can be performed and results produced.[4]

Evaluation

The development of ontology-enabled systems has some specific steps or considerations, noted above. However, the evaluation of built systems should follow best practices for evaluation of information systems in general.

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 a new 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 iteration 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 pre-deployment testing typically iterates until results indicate that it is safe to deploy the ontology

[4] For more details, see: http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Extrinsic_Aspects_Of_Ontology_Evaluation_Synthesis

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 operation environment, released, and becomes available for live use.

Evaluation

- 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 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?

Operation and Maintenance Phase

This phase involves the sustainment of deployed capabilities, rather than the development of new ones. A particular system may have ontology maintenance and new ontology development phases going on at the same time, but these activities should be distinguished as 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 a maintenance phase, information is collected about the results of operational use of the ontology. Any 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 use phase should not be regarded as part of maintenance phase; rather, they are inputs to, or part of, exploration activities 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 exploration and new development) during a use phase, 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 phase consists of identifying and addressing bugs or problems.

Evaluation

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, how 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 with the ontology, can it be corrected before the next major development and deployment cycle? If so, what is being done to address it?
- 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?

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 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.
Generally, appreciation of the full life cycle of an ontology is not well established within the ontology community. Thus, there are no tools for ontology development or to enable ontology evaluation across the whole life cycle. Existing tools support different parts of the ontology life cycle, and for any given characteristic, some tools may perform better in one life cycle phase than in another phase where a different tool is better suited.

Significant new ontology evaluation tools are currently becoming available to users. An overview will be presented as part of the Ontology Quality Software Survey (add link). Carving a link between such tools and existing IT architecture and design tools (e.g., Enterprise Architect and Solution Architect) 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.[5]

Observations and Recommendations

- We need to achieve a better understanding of the relationships between requirements at different levels and how low level requirements support higher level requirements. In particular, how business requirements translate into requirements for the ontology.
- Ontology development shares strong similarities with information systems development, and, thus, similar methodologies apply. Regardless how one conceptualizes the ontology life cycle (this document presents one possibility), requirements gathering is essential; without explicit requirements it is not possible to evaluate whether the ontology will support its intended use. Further, because the different levels of ontology (informal model, reference ontology, and implementation ontology) are evaluated against different kind of criteria, it is important to distinguish between ontologies of at those levels in order to evaluate them effectively.
- Evaluation methods and tools should be accompanied by explicit information about how the results of the evaluation relate to ontology requirements and during which life cycle phase(s) they are intended to be used. With respect to some results and metrics, tool developers themselves may be unclear on these relationships. Such uncertainty indicates needed research, to determine whether and how those results and metric are meaningful.
- There is a lack of tools that support the tracking of requirements and ontology evaluation across the whole life cycle of an ontology.
- Although there is much research on ontology evaluation and many organizations use sophisticated ontology evaluation and quality management practices, awareness of this research, these practices, and their importance to successful use of ontologies is neither widespread nor sufficiently pooled to constituted an accessible body of knowledge. Consequently, there is a lack of adoption of ontology evaluation techniques by the broader community of ontology consumers and developers. Ontology evaluation should be part of any ontology development, ontology usage or application. Ontology evaluation should happen across the whole life cycle of an ontology.

[5] For more details, see: http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Software_Environments_For_Evaluating_Ontologies_Synthesis

[[6]] *This final footnote is useful only if the summit community helps build out the library in the coming weeks.*

[6] Recommend resources with further detail concerning topics discussed in this document are collected, along with reference material consulted during the 2013 Ontology Summit, at https://www.zotero.org/groups/ontologysummit2013/items