- 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?
- Are naming conventions specified and, where names are provided, followed?
- Does the design call for multiple, distinct ontology modules? If so, do the ontology modules together cover the whole scope of the ontology?
- Does the design specify whether and how existing ontologies will be reused?
- Are all modules of the ontology associated with (informal) competency questions?
- 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 avoid addition of features or content not relevant to satisfaction of the requirements?

8. 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, 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, 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?

9. 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.

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?

Formalization of Competency Questions

Based on the results of the informal modelling, 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?

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 modelling 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.), 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 techniques 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]

---

[10] See https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/6GGPKU3D for more on expert evaluation of ontologies.
[11] See https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/929KF23Z for more on evaluating fidelity, including evaluation via simulation.
[12] For more details, see:
http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Intrinsic_Aspects_Of_Ontology_Evaluation_Synthesis and

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.

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 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)?

10. 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

---

http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Intrinsic_Aspects_Of_Ontology_Evaluation_CommunityInput 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?

11. 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 capability 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?

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

12. 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?

13. 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]

---

[14] See https://www.zotero.org/groups/ontologysummit2013/items/collectionKey/DWNMSJ5S & http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologySummit2013_Software_Environments_For_Evaluating_Ontologies_Synthesis, for more information about available tools.

[15] For example. For Survey results, see http://ontolog-02.cim3.net/wiki/Category:OntologySummit2013_Survey.

14. 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.