

Modelling Ontology Evaluation and Validation

Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann

Laboratory for Applied Ontology, ISTC-CNR, Roma, Italy
{aldo.gangemi, carola.catenacci, m.ciaramita,
jos.lehmann}@istc.cnr.it

Abstract. We present a comprehensive approach to ontology evaluation and validation, which have become a crucial problem for the development of semantic technologies. Existing evaluation methods are integrated into one single framework by means of a formal model. This model consists, firstly, of a meta-ontology called O^2 , that characterises ontologies as semiotic objects. Based on O^2 and an analysis of existing methodologies, we identify three main types of measures for evaluation: structural measures, that are typical of ontologies represented as graphs; functional measures, that are related to the intended use of an ontology and of its components; and usability-profiling measures, that depend on the level of annotation of the considered ontology. The meta-ontology is then complemented with an ontology of ontology validation called *oQual*, which provides the means to devise the best set of criteria for choosing an ontology over others in the context of a given project. Finally, we provide a small example of how to apply *oQual*-derived criteria to a validation case.

1 Introduction

The need for evaluation methodologies in the field of ontology development and reuse emerged as soon as 1994 (see [21]) and it has grown steadily ever since. Yet, no comprehensive and global approach to this problem has been proposed to date. This situation may become a serious obstacle for the success of semantic technologies, especially in industrial and commercial sectors. A typical example in this sense is the development of the Semantic Web. On the one hand, the idea of conveying semantics through ontologies definitely arouses the interest of large parts of the ICT Industry. Ontologies promise to be crucial components of web-like technologies that are able to cope with high interconnection, constant change and incompleteness. On the other hand, however, the lack of well-understood and shared notions of ontology evaluation and validation significantly slows down the transition of ontologies from esoteric symbolic structures to reliable industrial components.

In this paper we look at existing ontology-evaluation methods from the perspective of their integration into one single framework. To this end, we set up a formal model for ontology evaluation that consists, in the first place, of a meta-ontology – called O^2 – which characterises ontologies as semiotic objects. O^2 is meant to provide a foundation to the elements and features that are targeted by evaluation. Secondly, based on O^2 and an overview of the state of the art (cf. [8]), we provide a provisional catalogue of qualitative and quantitative measures for evaluating ontologies. We

identify three main types of measures: structural measures, that are typical of ontologies represented as graphs; functional measures, that are related to the intended use of an ontology and of its components, i.e. their function; usability-profiling measures, that depend on the level of annotation of the considered ontology. Thirdly, the meta-ontology is complemented with an ontology of ontology validation – *oQual* – which allows to pick up ontology elements by means of O^2 , provides quality-parameters and, when feasible, their ordering functions. Both O^2 and *oQual* are partly formalized in FOL and are currently maintained as OWL models, plugged into the DOLCE ontology library and its design patterns [25]. In practice, we model ontology evaluation as a diagnostic task based on ontology descriptions. Such descriptions make explicit some knowledge items that are crucial to ontology validation, like e.g.: roles and functions of the elements of the considered ontology; parameters for the descriptions that typically denote the quality of an ontology; and functions that compose those parameters according to a preferential ordering. At the end of the paper, we sketch an analytic examples of the trade-offs needed when composing principles with conflicting parameters, i.e. an application of *oQual*-derived criteria to a validation case. Finally, some conclusions are drawn.

2 O^2 : A Semiotic Meta-ontology

The use of meta-ontologies is becoming relevant within the semantic web, because of their easy integration and the shared construction methods with ontologies proper.

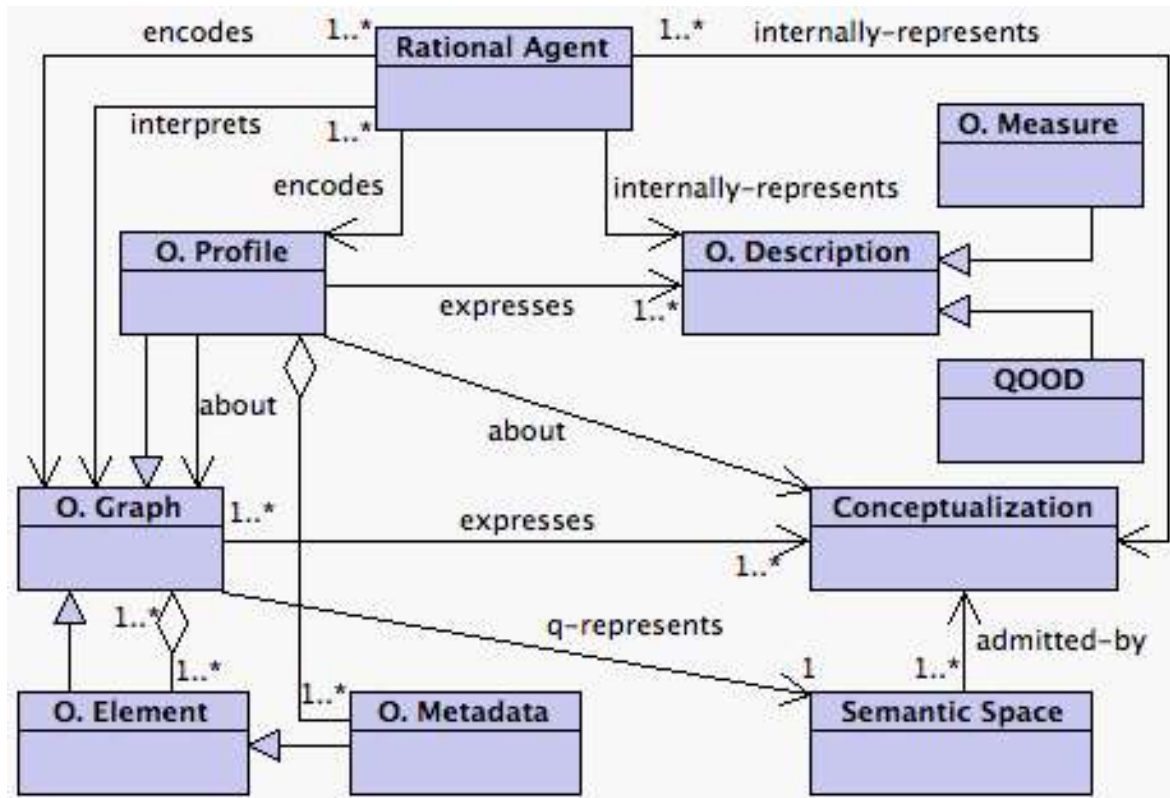


Fig. 1. A UML class diagram depicting the main notions from the O^2 ontology

[12], for instance, have recently proposed an *Ontology Metadata Vocabulary*. Following this tendency, we characterize an ontology as a semiotic object, i.e. an object constituted by an *information object* and an *intended conceptualization* established within a *communication setting*. The basic intuition behind this part of our proposal is that information can be constituted by any pattern that is used to represent another pattern, whereas that representation is interpretable by some rational agent as an explanation, an instruction, a command, etc. This is an idea that goes back at least to Peirce (cf. [17]).

This intuition is formalized by applying an ontology design pattern called *Information \leftrightarrow Description* (see [7]), and originates a new pattern called O^2 (because it is a “meta-ontology”). O^2 , in turn, formalizes the following specification: a) an ontology is information of a special kind; b) its patterns are graph-like structures; c) they represent intended conceptualizations, i.e. internal representations (by a rational agent) of *entity types*. For example, it is perfectly possible to define an ontology for subways, but one will hardly consider the graph of the London Underground as an ontology - at most, the latter can be considered as a *model* of an appropriate subway ontology.

In O^2 (Fig.1), an ontology graph has an intended conceptualization and a formal semantic space admitted by the conceptualization. The graph and the conceptualization are ‘kept together’ by a rational agent who encodes/interprets the graph, while internally representing its intended conceptualization.

An agent can also provide a profile containing metadata that express a “description” of the ontology, e.g. a method to measure the structural or functional properties of an ontology graph, its resulting attributes, its possible quality criteria and values, as well as its lifecycle annotations, such as provenance and informal annotations. A good profile typically enhances or enforces the usability of an ontology.

3 Measures for Ontology Evaluation

The literature on ontology evaluation is fragmentary. Most approaches address more or less specific evaluation issues but often do it unsystematically. Only [11] tries to disentangle issues by providing a classification grid for ontology evaluation methods. Such grid allows to present methods in terms of answers to the following questions: what is the considered method/tool like (Structure)? Subordinately: what is its goal (Goal)? What functions are supported by it (Function)? At which stage of development of an ontology may it be applied (Application)? Furthermore, how useful is the method? Subordinately: for which type of users is it conceived (Users types: Knowledge Engineers, Project Managers, Application Users, Ontology Developers)? How relevant is it to practice (Usefulness)? How usable is it (Usability)? For which type of uses was it conceived in the first place (Use cases)?

Partly based on such grid, and on an analysis of the related most relevant literature (notably, [24], [14], [23], [19], [18], [9], [6], and [16]), we have devised a large amount of possible measurement methods for ontologies and framed them in the pattern provided by O^2 .

We introduce here the main distinctions among measure sets, and provide a few examples for some of them. For the full list of identified measures and the detailed state-of-the-art review, see [8].

3.1 Measure Types

As explained above, by ontology we mean a *semiotic object* including graph objects, formal semantic spaces, conceptualizations, and annotation profiles; therefore, we propose to measure ontologies relatively to three main dimensions: structural, functional, and usability-profiling.

The *structural* dimension of ontologies focuses on syntax and formal semantics, i.e. on ontologies represented as *graphs*. In this form, the topological, logical and meta-logical properties of an ontology can be measured by means of a context-free metric.

The *functional* dimension is related to the intended use of a given ontology and of its components, i.e. their function in a context. The focus is on the conceptualization specified by an ontology.

Finally, the *usability-profiling* dimension focuses on the ontology profile (annotations), which typically addresses the communication context of an ontology (i.e. its pragmatics).

Notice that those dimensions follow a partition into *logical types*: structurally, we look at an ontology as an (information) object; functionally, we look at it as a language (information object+intended conceptualization), and from the usability viewpoint, we look at its meta-language (the profile about the semiotic context of an ontology). Therefore, the dimension types correspond to the constituents of the O^2 pattern and heterogeneous measurement methods are needed.

3.2 Measuring the Structural Dimension

In our treatment of the structural dimension, the idea is to define a general function like the following: $M = \langle D, S, mp, c \rangle$, where dimension D is a graph property or concept we want to measure: the intensional counterpart of the metric space; the set of graph elements S is a collection of elements in the graph (which may be seen as the ontology structure); mp is a measurement procedure; and c is a coefficient of measurement error.

The value of M is a real number obtained by applying a measurement procedure mp for a dimension D to a set S of graph elements, modulo a coefficient c (if any), i.e. (with an operational semantics): $mp_{D,c,S} \xrightarrow{\text{yields}} m \in \mathfrak{R}$

Within the possible sets of graph elements, we have considered the following :

- The set of graph nodes G from a graph g , $G \subseteq S$
- The set of root nodes $ROO \subseteq G$, where the root nodes are those having no outgoing *is-a* arcs in a graph g .
- The set of leaf nodes $LEA \subseteq G$, where the leaf nodes are those having no ingoing *is-a* arcs in a graph g .
- The sets of sibling nodes $SIB_{j \in G}$ connected to a same node j in a graph g through *is-a* arcs.
- The set of paths P where $\forall j \in P \Rightarrow j \subseteq G$, where a path j is any sequence of directly connected nodes in a digraph g starting from a root node $x_{\in ROO}$ and ending at a leaf node $y_{\in LEA}$.

- The set of levels (“generations”) L where $\forall j \in L \Rightarrow j \subseteq G$, where a generation j is the set of all sibling node sets having the same distance from (one of) the root node(s) $r \in ROO$ of a digraph g .
- The sets of graph nodes $N_{j \in P}$ from a same path j in a digraph g
- The sets of graph nodes $N_{j \in L}$ from a same level j in a digraph g
- The set MO of modules from a graph g . A *module* is any subgraph sg of g , where the set of graph elements S' from sg is such that $S' \subseteq S$. Two modules sg_1 and sg_2 are *taxonomically disjoint* when only ≥ 0 is-a arcs a_i connect sg_1 to sg_2 , and each a_i has the same direction.

Several structural measures can be defined, involving:

- a) topological properties such as *depth* (related to the cardinality of paths in a graph), *breadth* (related to the cardinality of paths in a graph), *tangledness* (related to multihierarchical nodes of a graph), and *fan-outness* (related to the ‘dispersion’ of graph nodes), among others;
- b) logical-adequacy properties such *consistency*, *anonymous classes* and *cycle ratios*, among others;
- c) metalogical-adequacy properties, such as e.g. *qualified density* (i.e. presence of meaningful conceptual-relation ‘dense’ areas, or ‘patterns’).

For instance, we have defined *depth* (a topological property) as a graph property related to the cardinality of paths in a graph, where the arcs considered are only is-a arcs. This measure only applies to digraphs (directed graphs). E.g., **average depth**, where $N_{j \in P}$ is the cardinality of each path j from the set of paths P in a graph g , and

$$n_{P \subseteq g} \text{ is the cardinality of } P: m = \frac{1}{n_{P \subseteq g}} \sum_j^P N_{j \in P}$$

3.3 Measuring the Functional Dimension

The functional dimension is coincident with the main purpose of an ontology, i.e. specifying a given conceptualization, or a set of contextual assumptions about an area of interest. Such specifications, however, are always approximate, since the relationship between an ontology and a conceptualization is always dependent on a rational agent that conceives that conceptualization (the ‘cognitive’ semantics) and on the semantic space that formally encodes that conceptualization (the ‘formal’ semantics) (Fig. 1). Hence, an appropriate evaluation strategy should involve a measurement of the degree of how those dependencies are implemented. We call this the *matching problem*.

The matching problem requires us to find ways of measuring the extent to which an ontology mirrors a given expertise (cf. [20]), competency (cf. [22]), or task: something that is in the experience of a given community and that includes not only a corpus of documents, but also theories, practices and know-how that are not necessarily represented in their entirety in the available documents. This seems to imply that no automatised method will ever suffice and that intellectual judgement

will always be needed. However, automatic and semi-automatic techniques can be applied that make evaluation easier, less subjective, more complete and faster [6].

The functional measures provided in [8] are variants of the *precision*, *coverage*, and *accuracy* measures introduced by [10], which are in turn based on an analogy with the *precision* and *recall* measures widely used in information retrieval (cf. [2]). They include *competence adequacy* (e.g. inter-subjective agreement, task adequacy, task specificity, and topic specificity); *NLP adequacy* (e.g. compliance with lexical distinctions), and *functional modularity* (e.g. ontology stratification, or granularity).

Due to the matching problem, however, the adaptation of precision, recall and accuracy to ontology evaluation is by no means straightforward. Since expertise is by default in the cognitive “black-box” of rational agents, ontology engineers have to elicit it from agents, or they can assume a set of data as a *qualified expression* of expertise and tasks, e.g. texts, pictures, diagrams, database records, terminologies, metadata schemas, etc. Therefore, we distinguish between *black-box* and *glass-box* measurement methods.

Black-box methods require rational agents, because they don't explicitly use knowledge of the internal structure of an expertise (see [8] for a more extensive discussion of these methods).

Glass-box methods require a data set that ‘samples’ that knowledge, and, on this basis, we can treat the internal structure of those data *as if* it were the internal structure of an expertise.

Based on these assumptions, precision, recall and accuracy of an ontology can be measured against: *a*) experts’ judgment, or *b*) a data set assumed as a qualified expression of experts’ judgment:

- (1) *Agreement (black-box)*: it is measured through the proportion of *agreement* that experts have with respect to ontology elements; when a group of experts is considered, we may also want to measure the *consensus* reached by the group’s members.
- (2) *User-satisfaction (black-box)*: it can be measured by means of dedicated polls, or by means of provenance, popularity, and trust assessment.
- (3) *Task*: what has to be supported by an ontology? (*glass-box*). It deals with measuring an ontology according to its fitness to some goals, preconditions, postconditions, constraints, options, etc. This makes the measurement very reliable at design-time, while it needs a reassessment at reuse-time.
- (4) *Topic*: what are the boundaries of the knowledge domain addressed by an ontology? (*glass-box*). It deals with measuring an ontology according to its fitness to an existing knowledge repository. This makes the measurement reliable both at design-time, and at reuse-time, but is based on the availability of data that can be safely assumed as related to the (supposed) topic covered by an ontology. Natural Language Processing (NLP)-based methods fit into this category, and are currently the most reliable method for ontology evaluation, at least for lightweight ontologies.
- (5) *Modularity*: what are the building blocks for the design of an ontology? (*glass-box*). It is based on the availability of data about the *design* of an ontology. Therefore, it deals with measuring an ontology according to its fitness to an existing repository of reusable components. This makes the measurement very reliable both at design-time, and at reuse-time. On the other hand, modularity

assessment is only practicable on ontologies that have been designed with an appropriate methodology.

As example of (*glass-box*) NLP-based measurements, consider a case in which the ontology is lexicalized (i.e., it defines, at least to some extent, what instances of classes and relations are called in natural language) and there exists a substantial amount of textual documents that contain information about the content of the ontology. By identifying mentions of ontological elements in a given corpus, it is possible to count the frequency of classes (or relations). The relative frequency of each class c (or relation r) is the proportion of mentions of ontology instances which are equal to c ; i.e., $P(c) = \text{count}(c) / \sum_i \text{count}(c_i)$. The relative frequency measures the importance of each class and provides a first simple measure of the ontology quality. For example, in newswire texts the three typical classes of ‘person’, ‘location’ and ‘organisation’ have somewhat similar frequencies, while if the corpus analysis reveals that one of the classes is much more unlikely than the others this means that there is something wrong with the instances of that class. This might indicate that the low frequency class is underrepresented in the ontology, at the lexical level. Recent work has focused also on discovering class attributes and arbitrary relation between classes through automatic or semi-automatic population of ontology objects (see [1], [5], and [4]). In fact, it is possible that new senses of already known instances are discovered, for example because the instance is polysemous/ambiguous (e.g., ‘Washington’ is both a person and a location).

3.4 Measuring the Usability Profile of Ontologies

Usability-profiling measures focus on the ontology profile, which typically addresses the communication context of an ontology (i.e. its pragmatics). An ontology profile is a set of ontology annotations: the metadata about an ontology and its elements. *Presence*, *amount*, *completeness*, and *reliability* are the usability measures ranging on annotations, which have been singled out in our research.

Annotations contain information about structural, functional, or user-oriented properties of an ontology. There are also purely lifecycle-oriented properties, e.g. authorship, price, versioning, organisational deployment, interfacing, etc. Three basic levels of usability profiling have been singled out: *recognition*, *efficiency*, and *interfacing*.

The *recognition level* makes objects, actions, and options visible (cf. [13]). Users need an easy access to the instructions for using ontologies in an effective way, and an efficient process to retrieve appropriate meta-information. That is, “give your users the information that they need and allow them to pick what they want”. Hence recognition is about having a complete documentation and to be sure to guarantee an effective access.

The *efficiency level* includes organisational, commercial, and developmental annotations. Large organisations tend to be compartmentalized, with each group looking out for its own interests, sometimes to the detriment of the organisation as a whole. Information resource departments often fall into the trap of creating or adopting ontologies that result in increased efficiency and lowered costs for the information resources department, but only at the cost of lowered productivity for the company as a whole. This managing-operating-balance principle translates into some

requisites (*parameters*) for the *organisation-oriented design* of ontology libraries (or of distributed ontologies), which provide constraints to one or more of the following entities: *organisation architecture*, *(complex) application middleware*, *trading properties*, *cost*, *accessibility*, *development effort*.

The *interfacing level* concerns the process of matching an ontology to a user interface. As far as evaluation is concerned, we are only interested in the case when an ontology includes annotations to interfacing operations. For example, a *contract negotiation* ontology might contain annotations to allow an implementation of e.g. a *visual contract modelling language*. If such annotations exist, it is indeed an advantage for ontologies that are tightly bound to a certain (computational) service. On the other hand, such annotations may result unnecessary in those cases where an interface language exists that maps to the core elements of a core ontology e.g. for contract negotiation.

4 *oQual*: A Model of Ontology Validation

We model ontology validation as a *diagnostic task* over ontology elements, processes, and attributes (Fig. 2). This task involves:

- *Quality-Oriented Ontology Descriptions (goods)*, which are a type of *ontology description* (cf. Fig.1) that provide the *roles* and *tasks* of, respectively, the elements and processes from/on an ontology, and have elementary goods (called *principles*) as parts. For example, a type of good is *retrieve*, which formalizes the requirement to be able to answer a certain competency question. In Fig. 2, the *retrieve* type is instantiated as a requirement for the ontology to be able to retrieve the ‘family history for a condition related to blood cancer’, in an ontology project for ‘blood cancer information service’.
- *Value spaces* (“attributes”) of ontology elements. For example, the presence of a relation such as: $R(p,f,c,i)$, where $Patient(p)$, $Family(f)$, $Condition(c)$, $Indicator(i)$.
- *Principles* for assessing the ontology fitness, which are modelled as elementary goods, and are typically parts of a project-oriented good. For example, ‘description of fitness to expertise’ is a principle.
- *Parameters* (ranging over the attributes -*value spaces*- of ontologies or ontology elements), defined within a principle. For example, ‘relation fitness to competency question’ is a parameter for the relation $R(p,f,c,i)$.
- *Parameter dependencies* occurring across principles because of the interdependencies between the value spaces of the measured ontology elements. For example, the ‘relation fitness to competency question’ parameter is dependent on either ‘first-order expressiveness’ or ‘presence of a relation reification method’ parameters ranging on the logical language of the ontology, because the relation $R(p,f,c,i)$ has four arguments and it is not straightforwardly expressible in e.g. OWL(DL).
- *Preferential ordering* functions that compose parameters from different principles. For example, in a ‘blood cancer information service’ project, the ‘relation fitness to competency question’ parameter may be composed with the ‘computational complexity’ parameter.

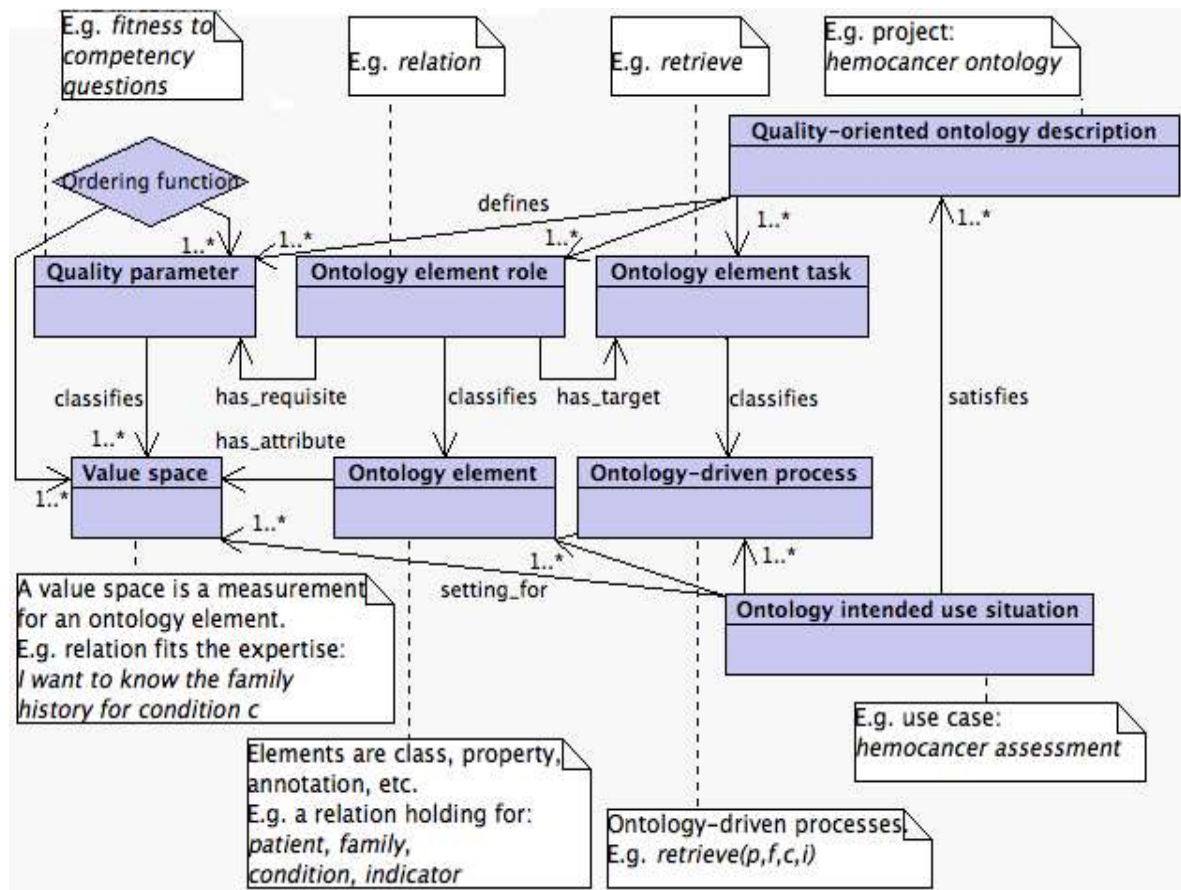


Fig. 2. The *oQual* design pattern applied to a clinical use case. A *good* based on the *fitness* to competency questions constrains the setting of the intended use for the ontology to be designed in a clinical information project.

- *Trade-offs*, which provide a conflict resolution description when combining principles with conflicting parameters. For example, the two abovementioned parameters might be conflicting when the cost of the expressiveness or of the reification method are too high in terms of computational efficiency. A trade-off in this case describes a guideline to simplify the competency question, or a strategy to implement the relation differently.

The *oQual* formal model is based on the the *Description* \leftrightarrow *Situation* pattern (cf. [7]) from the DOLCE ontology library ([15]), which is integrated with the *Information* \leftrightarrow *Description* pattern used for *O*².

Ontology descriptions, roles, parameters, and ordering functions are defined on the results of the measurement types that can be performed on an ontology graph, conceptualization, or profile. The results are represented as regions within value spaces. Quality parameters constrain those regions *within* a particular *good*.

5 Applying a *Good* to a Validation Case

In order to apply *oQual* to an analytic case of trade-off, we need a more detailed presentation of principles and of some of their typical parameters.

5.1 Some Principles and Parameters

Principles are defined here as structured descriptions of the quality of an ontology (*goods*): they are considered *elementary* goods because they usually define a limited set of parameters constraining ontology properties in order to support a common *goal*. Principles should also *lack conflicting parameters*.

Here is a list of some goods emerged in the practice of ontology engineering:

- Cognitive ergonomics
- Transparency (explicitness of organizing principles)
- Computational integrity and efficiency
- Meta-level integrity
- Flexibility (context-boundedness)
- Compliance to expertise
- Compliance to procedures for extension, integration, adaptation, etc.
- Generic accessibility (computational as well as commercial)
- Organizational fitness

The parameters defined by principles can be complex, but at the current state of research, they are usually simple scalars ranging on the measurement value spaces associated with the measures mentioned in Section 3.

Here is a list of parameters defined by the principles introduced above. For an easier understanding, each parameter is presented with the name of the measure on which it ranges, preceded by a + or – sign to indicate the scalar region constrained within the value space:

Cognitive ergonomics. Intuition: this principle prospects an ontology that can be easily understood, manipulated, and exploited by final users. Parameters:

-depth	-breadth
-tangledness	+class/property ratio
+annotations (esp. lexical, glosses, topic)	-anonymous classes
+interfacing	+patterns (dense areas)

Transparency. Intuition: this principle prospects an ontology that can be analyzed in detail, with a rich formalization of conceptual choices and motivations. Parameters:

+modularity	+axiom/class ratio
+patterns	+specific differences
+partitioning	+accuracy
+complexity	+anonymous classes
+modularity design	

Computational integrity and efficiency. Intuition: this principle prospects an ontology that can be successfully/easily processed by a reasoner (inference engine, classifier, etc.). Parameters:

+logical consistency	+disjointness ratio
-tangledness	-restrictions
-cycles	

A preferential ordering can either define the *prevalence* of a set of parameters from a principle p_1 over another principle p_2 , or it can define a composition of the two sets of parameters from p_1 and p_2 . A composition is the result of a *trade-off*. Both prevalence and trade-off descriptions are based on meta-parameters, e.g.: *available resources, available expertise, business relations, tools*, etc.

A simple exemplification of a trade-off for principle composition is the following. *Transparency* and *compliance to expertise* principles usually require *content ontology design patterns* (cf. [7]), involving *hub nodes* (classes with several properties, cf. [16]), then those principles require a *high rate of dense areas* parameter. But dense areas often need the definition of sets of (usually existential) axioms that potentially induce complex (in)direct cycles. Consequently, *high rate of dense areas* depends on a *high complexity* parameter (cf. [3] for the complexity of description logic ports of UML models).

The content design pattern for the *LimitViolation* pattern is an example of such a case (Fig.3). The *LimitViolation* pattern contains the following axioms (restrictions) that constitute a cyclical path, encoded here in OWL abstract syntax (corresponding to the red path in Fig.3):

```
Class(LimitViolation partial restriction(defines
someValuesFrom(ViolationParameter)))

Class(ViolationParameter partial restriction(classifies
someValuesFrom(ValueRegion)))

Class(ValueRegion partial restriction(observedBy
allValuesFrom(LegalControlSystem)))

Class(LegalControlSystem partial restriction(classifiedBy
someValuesFrom(LegalRole)))

Class(LegalRole partial restriction(d-used-by
someValuesFrom(LimitViolation)))

Class(LimitViolation partial restriction(defines
someValuesFrom(ViolationParameter)))

Class(ViolationParameter partial restriction(classifies
someValuesFrom(ValueRegion)))

Class(ValueRegion partial restriction(observedBy
allValuesFrom(LegalControlSystem)))

Class(LegalControlSystem partial restriction(classifiedBy
allValuesFrom(LegalRole)))

Class(LegalRole partial restriction(d-used-by
someValuesFrom(LimitViolation)))
```

If an ontology project using the limit violation axioms is based on a *good* that aims at both a *transparency* principle, and a *computational efficiency* principle, and we already know (cf. [8]) that computationally efficiency requires a *low rate of cycles parameter*, then we get a conflict of parameters (Fig.4). Therefore, a trade-off may be needed in an ontology project that uses the limit violation axioms. The trade-off can be applied by following two approaches.

The first approach defines a *preference ordering* over the parameters, which in the example leads either to accept the complexity, or to dismiss the pattern. The pattern is in this case essential to the ontology, then, if the *low rate of cycles* is also required because of e.g. available computational resources, we must resort to the second approach: *relaxation of parameters*. The possible methods to relax the parameters should act on either the reasoning algorithm, or the axioms. Since the first cannot be

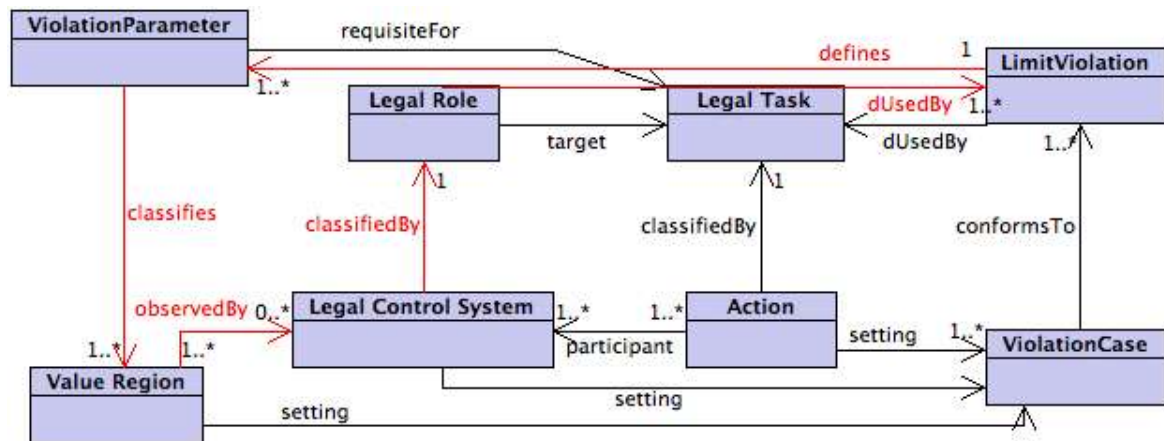


Fig. 3. The *LimitViolation* pattern in UML, showing a potential indirect cycle: a description of limit violation defines violation parameters ranging on some value space (e.g., speed), also assigning (legal) roles and tasks to legally-relevant entities: control systems, vehicles, persons, actions, etc. A violation case conforms to the description if legally-relevant entities and values are classified by parameters, roles, and tasks.

changed easily in most ontology projects, the best practice is to modify the model according to some *tuning practices* e.g. involving generalization over restrictions, which in our example can be done on one of the following axioms by substituting the class in the restriction with its superclass:

```
Class(ValueRegion partial restriction(observedBy
allValuesFrom(ControlSystem)))
```

```
Class(LegalControlSystem partial restriction(classifiedBy
allValuesFrom(Role)))
```

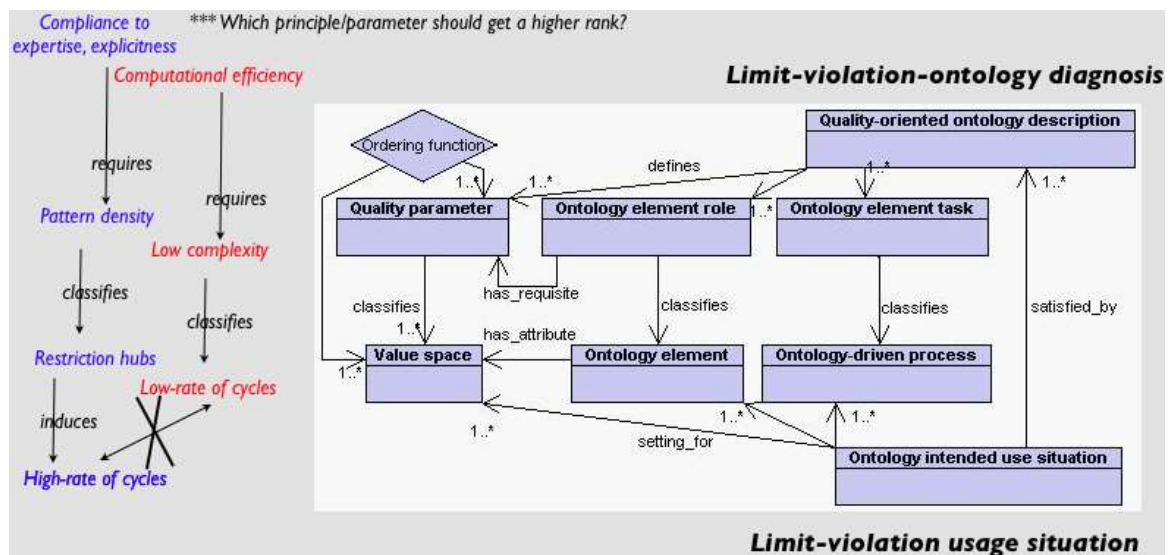


Fig. 4. A *qood* (a diagnosis of an ontology project using the limit violation pattern) that composes two principles requiring conflicting parameters

6 Conclusions and Future Work

O^2 and *oQual* are ontologies that characterise an ontology as a communication object, and allow to make a parametric design of evaluation and validation (diagnostic) tasks. Ontologies are analyzed in their graph and formal elements, functional requirements, and annotation profile. Therefore our approach results in parametric design specifications that address varied measures, ranging from graph properties to logical consistency, precision/recall, intersubjective reliability, etc., which do not suggest *prescriptive* validation of an ontology, but suggest an interactive, distributed validation against well-understood tasks.

Current work is focusing on the empirical assessment of the O^2 and *oQual* ontologies and related methods, by measuring existing ontologies, comparing the quality of distinct ontologies that represent the same domain, creating correlations between user-oriented and structural measures, and creating tools to assist ontology evaluation in large industry- and organization-scale projects (until now, the ontology and the method have been tested on fragments of large ontologies, and only thoroughly in the context of the Italian OntoDev project, featuring a mid-size lightweight ontology repository). Collaboration with the Oyster, Onthology, and KnowledgeZone projects ([26]) are being established in order to harmonize ontology metadata semantics with tools, and to include/extract evaluation annotation in/from current metadata vocabularies.

References

1. Almuhareb A. and Poesio M., (2004), "Attribute-based and value-based clustering: an evaluation", in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
2. Baeza-Yates R. and Ribeiro-Neto B., 1999, *Modern Information Retrieval*, Addison Wesley.
3. Berardi D., Calvanese D., De Giacomo G., (2001), "Reasoning on UML Class Diagrams using Description Logic Based Systems", *Proceedings of the KI'2001 Workshop on Applications of Description Logics*.
4. Brewster C., Alani H., Dasmahapatra S. and Wilks Y., (2004), "Data-driven ontology evaluation", in *Proceedings of LREC*.
5. Ciaramita M., Gangemi A., Ratsch E., Saric J., and Rojas I., (2005), "Unsupervised Learning of Semantic Relations between Concepts of a Molecular Biology Ontology", in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*.
6. Daelemans W. and Reinberger M.L., (2004), "Shallow Text Understanding for Ontology Content Evaluation", *IEEE Intelligent Systems*: 1541-1672.
7. Gangemi A., (2005), "Ontology Design Patterns for Semantic Web Content", in Motta E. and Gil Y. (eds.), in *Proceedings of the Fourth International Semantic Web Conference*.
8. Gangemi A., Catenacci C., Ciaramita M., and Lehmann J., (2005), "Ontology evaluation: A review of methods and an integrated model for the quality diagnostic task", *Technical Report*, available at <http://www.loa-cnr.it/Publications.html>.
9. Gómez-Pérez A., (2003), "Ontology Evaluation", in *Handbook on Ontologies*, S. Staab and R. Studer (eds.), Springer-Verlag, pp. 251–274.

10. Guarino N., (2004), "Towards a Formal Evaluation of Ontology Quality", *IEEE Intelligent Systems*: 1541-1672.
11. Hartmann J., Spyns P., Giboin A., Maynard D., Cuel R., Suárez-Figueroa M.C., and Sure Y., (2004), "Methods for ontology evaluation", *Knowledge Web Deliverable D1.2.3*,
12. Hartmann J., Palma R., Sure Y., Suárez-Figueroa M.C., and Haase P. (2005), "OMV—Ontology Metadata Vocabulary", paper presented at the *Ontology Patterns for the Semantic Web (OPSW) Workshop at ISWC2005*, Galway, Ireland:
<http://www.research.ibm.com/people/w/welty/OPSW-05/>.
13. Kaakinen, J., Hyona, J., and Keenan, J.M., (2002), "Individual differences in perspective effects on on-line text processing", *Discourse Processes*, 33: 159 - 173.
14. Lozano-Tello, A. and Gómez-Pérez A., (2004), "ONTOMETRIC: A method to choose the appropriate ontology", *Journal. of Database Management*, 15(2).
15. Masolo, C., A. Gangemi, N. Guarino, A. Oltramari and L. Schneider, (2004), "WonderWeb Deliverable D18: The WonderWeb Library of Foundational Ontologies", available at <http://www.loa-cnr.it/Publications.html>.
16. Noy, N., (2004), "Evaluation by Ontology Consumers", *IEEE Intelligent Systems*: 1541-1672.
17. Peirce C.S., (1931-1958), *Collected Papers*, vols. 1-8, C. Hartshorne, P. Weiss and A.W. Burks (eds), Cambridge, MA: Harvard University Press.
18. Porzel R. and Malaka R., (2004), "A Task-based Approach for Ontology Evaluation", in *Proceedings of ECAI04*.
19. Spyns P., (2005), "EvaLexon: Assessing triples mined from texts", *Technical Report 09*, STAR Lab, Brussel.
20. Steels L., (1990), "Components of Expertise", *AI Magazine*, 11, 2: 30-49.
21. Sure Y. (ed.), (2004), "Why Evaluate Ontology Technologies? Because It Works!", *IEEE Intelligent Systems*: 1541-1672.
22. Uschold U. and Gruninger M., (1996), "Ontologies: Principles, Methods, and Applications," *Knowledge Eng. Rev.*, vol. 11, no. 2: 93–155.
23. Welty C. and Guarino N., (2001), "Supporting ontological analysis of taxonomic relationships", *Data and Knowledge Engineering*, vol. 39, no. 1, pp. 51-74.
24. Yao H., Orme A.M., and Etzkorn L., (2005), "Cohesion Metrics for Ontology Design and Application", *Journal of Computer Science*, 1(1): 107-113.
25. http://www.loa-cnr.org/ontologies/DLP_397.owl, <http://dolce.semanticweb.org> .
26. <http://oyster.ontoware.org>; <http://www.onthology.org>; <http://smi-protege.stanford.edu:8080/KnowledgeZone/>.