ON ESTABLISHING AN ONTOLOGY REENGINEERING FRAMEWORK

Dionysia Kontotasiou, Charalampos Bratsas and Panagiotis D. Bamidis
Medical Informatics Laboratory, School of Medicine, Aristotle University of Thessaloniki, Thessaloniki, Greece

Keywords: Knowledge engineering practices, Ontology evaluation frameworks.

Abstract: A set of ontology evaluation criteria are specified in this paper in order to ensure that existing ontologies adhere to a set of requirements in order to be reusable in various contexts. The proposed evaluation criteria are designed in principle to provide the means for the improvement of existing ontologies and the development of new ones with efficient structure, increased readability and limited redundancy. Existing ontologies play a useful role in the development of new ones, because authoring ontologies from scratch is a costly and non-trivial task. On the other hand, reusing existing ontologies may save significant effort and helps interacting with different development tools. Based on practical experience, as well as existing ontology evaluation methodologies, we propose a set of specifications that should be taken into account at any ontology authoring or restructuring process. On top of this, we define a set of evaluation metrics in order to quantitatively assess the improvement that is potentially achieved by the application of the refinement process. The generalization of the application of the proposed criteria on a large-scale basis is the next step to establish an integrated ontology evaluation framework.

1 INTRODUCTION

In the context of knowledge engineering and information sciences, ontologies define a set of representational primitives that model a domain of knowledge or discourse (Gruber, 2008). Building an ontology or an ontology network from scratch is not always an easy process. Even though many visualization support tools are available that facilitate the various steps of the ontology lifecycle, the core development of an ontology remains a manual task that requires good knowledge of the domain to be modeled, as well as good modeling skills and experience. It is a common practice for knowledge engineers to work together with domain experts in order to build robust ontologies.

This paper deals with the ontology refactoring process, which is part of an ontology authoring process, when an existing ontology is used as a basis. Moreover, ontology refactoring or refinement can be applied for the purpose of improving an existing ontology, according to a set of evaluation criteria. The reason, for which ontology refactoring or ontology evaluation has obtained noticeable interest in the last years, is that creating a new application-specific ontology from scratch is usually a time-consuming and cost-effective task by nature. On the other hand, reusing existing ontologies may save significant effort and helps interacting with different development tools.

It is a common practice, when a new ontology comes to describe a domain or to be used as part of an overall application, to consider reusing one or more of existing candidate ontologies already created for similar use (Borgida and Giunchiglia, 2007). In addition to this, several applications can use the same domain ontology to solve different problems, and the same problem-solving method can be used with different ontologies. However, this practice requires that the ontologies to be reused adhere to a set of informal specifications in terms of their vocabulary, syntax, structure, documentation, data formalisms, etc. When a candidate ontology fails to fulfill this requirement it is likely that an improvement, restructuring and in general refinement of the ontology is necessary in order to make the ontology more suitable for reuse. This paper identifies and groups a set of such specifications that constitute at the same time a set of guidelines for ontology development that can be applied with the goal to shape a preliminary ontology evaluation framework.

Kontotasiou D., Bratsas C. and D. Bamidis P..
ON ESTABLISHING AN ONTOLOGY REENGINEERING FRAMEWORK.
DOI: 10.5220/0003629003540360
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2011), pages 354-360
ISBN: 978-989-8425-80-5

In general, our work defines a set of ontology and evaluation criteria to be applied to existing ontologies for the purpose of their refactoring or evaluation.

2 ONTOLOGY REFINEMENT AND EVALUATION CRITERIA

This section analyzes the most important aspects to be considered during the restructuring phase as part of our evaluation methodology. The analysis that follows is based on the layer-oriented approach that was defined in the Ontology Summit 2007 (Gruninger et al., 2007). These layers provide a taxonomy of the identified ontology issues that should be taken into account during the refinement process. Each one of the identified issues or properties indicates a necessary step in the refinement and evaluation process that should be followed in order to improve the ontologies in their original form. The proposed layers or dimensions are distinguished between internal and external ones. Internal dimensions are concerned with the ontologies themselves, their internal organization, naming conventions, representation, and so on. The external measures are related to their take-up and use within user communities, their role as standards, embedding within business practices, and so on.

In particular, the basic internal dimensions or 'layers' are listed below:

- Lexical/Vocabulary layer – This layer includes all restructuring attributes that are relevant to the syntactic elements of ontologies, such as naming conventions.
- Structural/Architectural layer – It includes all aspects that characterize the structural attributes of ontologies, i.e., concept and property hierarchy, grouping of similar ontological concepts, that are repeated and removal of unused modules.
- Representational/Semantic layer – This layer relates to the semantic elements of ontologies, i.e., attributes whose goal is to conceptually describe the structural ontology elements, such as documentation and visualization.
- Data/Application layer – The fourth internal layer covers attributes relevant to how an ontology applies to a given domain. Domain range definition of properties is listed as an attribute of this layer.

In addition to the internal layers listed above, there is an external one:

- Usability layer – It includes quality measures that are required to ensure that the resulted ontologies satisfy a set of usability standards. Disjointness restrictions belong to this layer.

2.1 Naming Conventions

Naming conventions (Schober, 2009) refer to the way all elements of an ontology are named and belong to the lexical/vocabulary layer, because naming is basically part of the syntactic features of ontologies. It has to do with the formulation of "good" terms and definitions, where essential features should be satisfied by all naming conventions (e.g. nominal, verbal, etc). According to this criterion circularity in definitions should be avoided and "junk" categories should be eliminated.

As an example, there may be some concepts modeling similar kinds of information. These concepts usually begin with the same prefix and end with a different suffix, or inversely. However, it is often observed that not always the same prefix/suffix is used. In this case, these concepts should be aligned for reasons of clarification and clearness and follow the same naming conventions (e.g. begin with the same prefix or end with the same suffix). Furthermore, plural/singular forms and the use of camel-case or use of the underscore symbol should not be mixed.

2.2 Concept Hierarchy/Taxonomy

This aspect belongs to the structural/architectural layer, because hierarchies that are defined between concepts and properties determine the way in which the ontology will be structured. On the other hand, ontologies are formed as taxonomies that are built around concrete configurations of the different hierarchies amongst ontological elements. This criterion may be quantitatively evaluated by metrics such as the size, the depth, and the breadth of hierarchy, the density (average branching of concepts), etc, which provide a measure of the complexity of the overall taxonomy.

A flat concept hierarchy, for instance, usually implies that there are too many concepts on the same level. This indicates the existence of unexploited grouping possibilities for concepts with similar semantics; hence these concepts should be grouped together under one more general concept. Specifically, the problem with flat concept hierarchy is that everything exists everywhere at once and all on the same level. Thus, there is no modularity, openness or depth in these ontologies and there is a growing appreciation that ontologies are evolutionary. However, evolutionary theory demands a clear identification of variation, interaction and selection but a flat ontology can make no sense of this.

Another example is the existence of branches with different structures. This may result in too deep ontologies and unbalanced taxonomies. Finally, the level of abstraction, to which the concepts refer, is not always taken carefully into account, thus resulting in an inappropriate ontology structure.

All of the above issues need to be considered during the ontology restructuring phase. For instance, a flat concept hierarchy can be converted to a more arborescent (tree-like) structure, so as to reduce the number of concepts on the same level. Exploiting the grouping possibilities for concepts of similar kinds results in a better grouping and a more clear reorganized structure of the ontology. A more appropriate structure for ontologies can also be achieved by grouping together on the same hierarchy level all concepts that refer to the same level of abstraction. Finally, the structure of branches, which are very different than others can change in order to have a more balanced and equally developed hierarchy.

2.3 Property Related Issues

This category is composed of two refinement criteria: property structure and property restrictions. It also belongs to the structural/architectural layer of the ontology authoring process because hierarchy applies to properties in a similar way that is applied to concepts.

2.3.1 Property Structure

Property structure may be quantitatively evaluated by similar metrics as in Section 2.2 such as the size, the depth/breadth of hierarchy, density and complexity of the hierarch of object and data properties. Issues that are addressed by this criterion include the lack of well structured properties in ontologies, when there is a clear hierarchical relationship between different properties that share common characteristics. The need for adding hierarchical relationships between properties occurs when properties are poorly gathered into conceptual groups of similar properties. In this case a restructuring process is deemed necessary by exploiting grouping possibilities for properties of equal domains/ranges or their functions. By introducing one or more levels of hierarchy between these properties we achieve a more efficient representation of the involved properties that also results in the reduction of redundant information within the definition of each property. On top of this, the application of restructuring processes to ontology properties can reduce the number of properties on the same level and produce a more hierarchical structure about properties. This implies a more concrete and understandable ontology structure.

2.3.2 Property Restrictions

We can use properties in order to create restrictions. This feature is common in the Web Ontology Language (OWL). As the name suggests, restrictions are used to impose various restrictions to the individuals that belong to a class. Restrictions in OWL fall into three main categories:

- Quantifier Restrictions: AllValues From ($\forall$), SomeValues From ($\exists$).
- hasValue Restrictions.
- Cardinality Restrictions.

Quantifier and *hasValue* constraints constitute restrictions on the kinds of values a property can take, while cardinality restrictions on the number of values a property can take. Property restrictions can easily be evaluated by the number of various restrictions that exist in an ontology.

The total time for checking ontology consistency depends on the size of the initial ontology but also on the use of these restrictions. Constructs like *SomeValuesFrom*, *MinCardinality*, and *MaxCardinality* will cause the consistency algorithm to create new nodes in the ontology. Applying this algorithm to new nodes will require more processing time. Thus, by deleting some of the existing restrictions we achieve a faster "check consistency mechanism" of the involved properties.

2.4 Grouping Similar Ontological Concepts

Also in the architectural layer we define a criterion about grouping similar concepts that appear in ontologies. This criterion is classified in the architectural layer as it deals with modularization issues, such as what modules are defined in the ontology, how they are defined, if they can be imported/exported/reused and so on, that have a primarily impact on the ontology structure. According to this criterion if similar ontological concepts are repeated frequently throughout the structure, they can possibly be combined to one module and reused whenever necessary. Hence, duplicate concepts can be defined only once and their use be extended within other definitions.

The implication of grouping similar ontological concepts in order to avoid their repetition is to make maintenance of the specified modules easier, e.g. it becomes a trivial task for ontology authors to add or remove something in the ontology or to keep track of the naming issues in general, because naming is preserved and this results in less typing errors. In any case, the definition of modules depends on the language to be used, what is intended to represent, and the applicability of reusing the modules.

2.5 Documentation/Visualization

The documentation and visualization criterion belongs to the representational/semantic ontological layer because it encompasses issues such as how the ontology is represented in the outside world and how it is described in terms of the semantics of its elements. In particular, this criterion addresses documentation and term governance, among others. It involves the activity of enriching the ontology with additional information, such as free text comments or annotations, metadata, implementation code and so on, as well as the collection of documents and explanatory comments generated during the entire ontology building process. In general, this aspect refers to anything that could be helpful to make the ontology more readable, to users for whom the ontology is intended.

Based on experience, it seems that documentation and visualization concerns are usually left as a final task by the ontology authors. Thus, ontologies are usually poorly documented, with few or almost no comments. This results in ontologies that even if they are consistent in terms of their syntax and semantics, they are difficult to use and understand, especially by those users who aim to apply or reuse them. In this case as this criterion dictates, the documentation and visualization aspects of an ontology should be improved and comments should be added for a better description and clarification of various ontology parts. After providing sufficient documentation to an ontology, it will become easier for this to be applied, reused, and consumed by other applications.

2.6 Disjointness Restrictions

Last but not least disjointness restrictions (Rector, 2003) mainly affect the usability layer of the ontology when it comes to be used as part of an overall application, e.g. when instances are added, forms are created, or queries have to be responded. These restrictions are applied on ontology classes or properties in order to apply limitations to the domain in which they are used. Thus, by properly defining classes and properties their usability is enhanced as their reuse by other applications is sufficiently enabled.

Although most concepts inside the ontology are usually pairwise disjoint with each other, this condition is sometimes missing for some concepts. On the other hand, for some other concepts disjointness might not hold, but where there might be an overlap. In such a case, if for example there may exist an individual that is an instance of two classes, disjointness restriction should be removed from these two classes.

In general, the issue of disjointness restrictions should be considered more carefully on ontology development or restructuring. That is, for concepts where it is necessary, the missing disjointness condition should be added. Similarly, for some other concepts where an overlap may occur and a specific individual may be an instance of all of them, disjointness does not hold and they should not be made pairwise disjoint with each other.

3 EVALUATION METRICS

Here we introduced specific ontology evaluation metrics that are derived from the previous criteria. Sections 3.1-3.6 describe the measurable metrics for each restructuring criterion of our ontology evaluation process.

3.1 Naming Conventions

In order to assess in a measurable way how well the naming conventions criteria are fulfilled by an existing ontology we introduce the following three metrics.

- *N1: Classes with the same naming conventions.* This metric is equal to the percentage of the majority of classes that adopt the same naming convention schema, such as camel-case notation, singular form of words and upper case letter. The value of this parameter ranges from 0%, when none of the classes adopt any naming convention standard, to 100% where all classes adopt the same standard. The value of this parameter indicates the extent to which the ontology adopts a common naming standard.
- *N2: Object properties with the same naming con-*

ventions. This metric is the same as the previous one but it applies on object properties instead of classes and takes into account property names that begin with a lower-case letter.

▪ *N3: Data-type properties with the same naming conventions.* Similarly, this metric is defined as in the previous case but it applies on data-type properties.

3.2 Concept Hierarchy/Taxonomy

Concept hierarchy expresses how well a specified taxonomy is structured. The measurable criteria that are used in order to assess this feature are associated with the number of classes, average number of parent and sibling nodes, as well as various metrics about the characteristics of the tree taxonomy, such as the tree depth, the internal and external paths, and so forth. The total list of these criteria follows.

▪ *C1: Total Number of Classes.* It is defined as the number of classes in the ontology.

▪ *C2: Number of Primitive Classes.* This metric equals the number of classes in the ontology that have necessary conditions. When necessary conditions are defined for a class, any instance of this class should necessarily fulfill these conditions. However, if any instance fulfils these conditions, this does not necessarily imply that it is also a member of this class.

▪ *C3: Number of Defined Classes.* It is equal to the number of classes in the ontology that have at least one set of necessary and sufficient conditions. When necessary and sufficient conditions apply to a class, any member, i.e., instance of this class should necessarily fulfill these conditions, and vice versa, if any instance fulfils these conditions then it is certainly a member of this class.

▪ *C4: Average Number of Parents.* This metric expresses the average number of parent classes, or "super-classes" based on each class in the taxonomy. The greater the value of this metric is, the denser the structure of the ontology becomes.

▪ *C5: Maximum Number of Parents.* Similarly to the previous metric, this one is equal to the maximum number of super-classes that correspond to all ontology classes. This is a structure-related metric that expresses the maximum number of isa hierarchy associations that are defined per class.

▪ *C6: Average Number of Siblings.* This metric is the average number of sibling classes, i.e., classes that share the same parent of all ontology classes. This metric expresses the average number of child nodes per hierarchical level per parent class. As the value of C6 increases, the ontology becomes denser, and the number of child nodes increases per parent node.

▪ *C7: Maximum Number of Siblings.* This metric displays the maximum number of classes that share the same parent node in the ontology. This is also a metric of how dense an ontology is in terms of its structure. A big value for C6 indicates a dense ontology with a big number of child nodes per parent node.

▪ *C8: Max Depth.* Given an ontology tree, this metric computes the maximum depth of the tree structure, namely the number of nodes along the longest path from the root node down to the farthest leaf node. This metric indicates the number of structure levels within the ontology. A big value for C8 indicates that the taxonomy consists of many hierarchy levels.

▪ *C9: Total Number of Nodes.* It is the total number of nodes in the ontology tree structure. This is a metric about how dense is the ontology structure.

▪ *C10: Total Number of Roots.* The total number of nodes that belong to the topmost level in the ontology tree hierarchy, i.e., the number of nodes with no parents. This indicates the number of independent classes that are defined within the same taxonomy. It is a measure of ontology modularity.

▪ *C11: Total Number of Internal Nodes (Parents).* It is equal to the total number of nodes in the ontology tree. Only nodes with child nodes are taken into account. This metric expresses how dense is the ontology structure.

▪ *C12: Total Number of Children.* It is equal to the total number of child nodes in the taxonomy, i.e., nodes with at least one parent node. This metric also expresses the density of the tree structure.

▪ *C13: Total Number of External Nodes (Leaf).* It is defined as the total number of nodes in the ontology tree structure that do not have any child nodes. Root nodes are also taken into account for the calculation of this metric. Again, this is a taxonomy-density metric.

▪ *C14: Internal Path Length.* It is equal to the sum over all internal nodes of the paths from the root of the taxonomy to each node, not including tree leaves, i.e., nodes with no children. The depth for an internal node is defined as the number of classes that we come across when traversing the tree from the root to the internal node.

▪ *C15: External Path Length.* This metric is defined as the sum over all external nodes, i.e., leaves, of the lengths of the paths from the root to each node. Both C14, C15 are metrics that express tree density.

3.3 Property Metrics

General property metrics are used to measure the total number of properties in the taxonomy as well as the total number of properties of each type (i.e., object, data-type and annotation properties). In particular, the following metrics are defined.

▪ *P1: Total Number of Properties.* This metric is equal to the total number of properties in the ontology (including object, data-type, and annotation properties). It holds that *P1 = P2 + P3 + P4.* Metrics P2, P3 and P4 are described below.

▪ *P2: Number of Object Properties.* It is equal to the number of object properties in the ontology. Object properties provide associations between individuals of the same or different classes in the ontology.

▪ *P3: Number of Data-type Properties.* Similarly, this metric is defined as the number of data-type properties that associate individuals to XML-schema data types or RDF literals.

▪ *P4: Number of Annotation Properties.* This metric counts the number of annotation properties. These properties are used for documentation purposes, such as to add metadata to classes, individuals and properties.

▪ *P5: Properties with an inverse specified.* It provides the number of properties for which an inverse property is specified.

▪ *P6: Total Number of Restrictions.* In OWL, properties are used to create restrictions. This metric is defined as the number of various restrictions that are imposed to individuals (instances) of a class. Restrictions in OWL fall into four main categories: existential, universal, cardinality and hasValue restrictions. Based on these, the following additional metrics P7 to P12 are defined.

▪ *P7: Number of Existential Restrictions.* This metric is equal to the total number of restrictions applied on individuals with at least one property from a specific range.

▪ *P8: Number of Universal Restrictions.* It is defined as the number of restrictions that are imposed on properties with exactly one range.

▪ *P9: Cardinality Restrictions.* In OWL, we can describe the class of individuals that have at least, at most or exactly a specified number of relationships with other individuals or data-type values. The restrictions that describe these classes are known as cardinality restrictions. This metric is equal to the number of such restrictions. There are two specific types of cardinality restrictions: *MinCardinality* and *MaxCardinality* that are described by metrics P10 and P11, respectively.

▪ *P10: MinCardinality Restrictions.* It is equal to the number of restrictions that impose a minimum number of relationships in which an individual is allowed to participate.

▪ *P11: MaxCardinality Restrictions.* It is equal to the number of restrictions that impose a maximum number of relationships in which an individual is allowed to participate.

▪ *P12: HasValue Restrictions.* This metric counts the number of *hasValue* restrictions that define an anonymous class of individuals as a range for a specific property. The *hasValue* restriction associates a specific property to a tangible entity (i.e., a string) that is assigned as a value to the property.

All of the above metrics express the extent to which the various properties in an ontology are imposed to restrictions. Restrictions indicate that special care has been taken on the concrete definition of ontology properties.

3.4 Grouping Similar Ontological Concepts

The reuse mechanism of ontological concepts can be evaluated directly from metrics G1, G2 that are defined below.

▪ *G1: Total Number of Similar Classes.* This metric provides the total number of similar classes in the ontologies and indicates the semantic duplicates that exist on them.

▪ *G2: Total Number of Similar Properties.* It is equal to the total number of similar properties. This metric indicates the extent of semantic duplicates regarding the properties in the ontology.

3.5 Documentation/Visualization

The goal of the documentation/visualization metrics is to assess the amount of information that is included in the ontology for documentation purposes. This information may be included in the various elements in the ontology as free text comments, annotations, or metadata that facilitate the understanding and reuse of the ontology elements by third-party practitioners. We define the following metrics:

▪ *D1: Total Number of Documented Classes.* This metric provides the total number of documented classes and it indicates the extent to which an ontology is documented. The higher the value of D1 becomes, the more documentation-related information is included in the ontology.

- *D2: Total Number of Documented Properties.* It is equal to the total number of documented properties. Similarly, this metric indicates the extent of documentation regarding the properties in the ontology.
- *P11: Number of Annotation Properties.* This metric has been defined in the property category because it is associated with both properties-related and documentation-related issues in an ontology. It is defined as previously, to be the total number of annotation properties occurring in the ontology. This type of properties is useful for writing metadata to classes, individuals and properties.

3.6 Disjointness Restrictions

The definition of disjointness restrictions on classes prevents those classes from overlapping with each other, thus creating confusion to reasoners. In order to specify the extent to which classes in an ontology are defined as disjoint, we introduce the metric J as the total number of disjointness restrictions on classes. Based on experience, since not all of the classes in an ontology should be disjoint, this metric is used to indicate whether such types of constraints are taken into account or not during the design of an ontology.

4 CONCLUSIONS

In this paper we presented a methodology whose goal is to provide a set of guidelines and indicate a best-practice approach for ontology re-structuring and refinement. The expected evolution of the presented methodology is to shape a formal ontology evaluation framework that can be applied in a two-fold way; firstly, as a set of guidelines and best practices for newly created ontologies, and secondly, as a formal ontology framework for existing ontologies.

In order to achieve this expectation, further work is required. Our future plans include the development of a supporting software framework with a set of tools that will automate the evaluation process, as much as possible. Moreover the provided tools will facilitate the evaluation process on behalf of ontology authors by the provision of appropriate user interface abstractions and facilities. On the other hand, further work is required in order to formalize the presented theoretical framework in the best possible way, so that it can form a proposal for either establishing a new standard on ontology evaluation methodologies, or contributing to existing relevant standardization efforts. In both ways, it is expected that our evaluation methodology will fulfill in the best possible way an existing and recognized need for a tangible and efficient ontology evaluation framework capable to be used on a large-scale basis.

REFERENCES

Gruber, T., 2008. Ontology. In Encyclopedia of Database Systems, *Springer*.

Borgida, A., Giunchiglia, F., 2007. Importing from Functional Knowledge Bases. In *Proceedings of 2nd International Workshop on Modular Ontologies* Whistler, Canada.

Gruninger, M., Bodenreider, O., Olken, F., Obrst, L., Yim, P., 2007. The 2007 Ontology Summit: Ontology, Taxonomy, Folksonomy: Understanding the Distinctions. *Journal of Applied Ontology*.

Schober, D., Smith, B., Lewis, S., Kusnierczyk, W., Lomax, J., Mungall, C., Taylor, C., Rocca-Serra P., Sansone, S. A., 2009. Survey-based naming conventions for use in OBO Foundry ontology development. *BMC Bioinformatics*.

Rector, A. L., 2003. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In *Proceedings of the 2nd international Conference on Knowledge Capture*.