Quality of an Ontology as a Dynamic Optimisation Problem

Michael Cochez and Vagan Terziyan

Department of Mathematical Information Technology
University of Jyväskylä
P.O. Box 35 (Agora), 40014, Jyväskylä, Finland
`firstname.lastname`@jyu.fi

Abstract. The Semantic Web is a proposal from the World Wide Web Consortium aimed at solving problems like data integration and application interoperability. To reach these goals several languages for the representation of semantic data have been proposed. One of the essential concepts behind semantic data is that the data is according to a certain ontology. However, the goals of the semantic web seem challenged because it seems essential for its working that ontologies are agreed upon and shared. This work-in-progress paper describes a first step to solving these problems. When an ontology is missing or only partially known a system might try to make an approximation of the missing part of the ontology. The quality of this estimated ontology will depend on the context of the application. This paper proposes the solving of a dynamic multi-objective and context sensitive optimisation problem as a way to evalute the quality of the ontology.

Keywords: Semantic Web, Ontology features, Ontology quality, Contextual optimisation
Key Terms: KnowledgeEvolution, Intelligence, SemanticWebService

1 Introduction

The World Wide Web Consortium (W3C) regards the Semantic Web as a web of data. Currently, data is created at extremely high rate and is not available enough to end users. The Semantic Web aims "To do for machine processable information (application data) what the World Wide Web has done for hypertext" by supporting the creation of interoperable and linked data. [1] Linking data mostly happens by using shared identifiers and linking concepts by using shared ontologies.

The paper "Which Semantic Web?" [2] gives quite a critical view on many concepts of the Semantic Web. It states for instance that "Agreeing on a cataloging scheme for Semantic Web documents is a prerequisite for any sharing of semantic knowledge. " and "It is easy enough for computers to exchange data about computational abstractions such as filenames, sizes, usernames, passwords, etc. It is much harder for computers to exchange information about human-oriented concepts such as happiness and beauty.”. These statements indicate that the Semantic Web might actually fail in its basic ideas of making decentralised information management possible. This work-in-progress paper describes the idea behind one possible approach to overcome these problems.

In order to address the first problem, it would be necessary to create a system which can make use of semantic data without having knowledge about the ontology used by the system which generated it. Therefore, it would be needed to derive the meaning from the data which another application generated. The approach proposed in this paper is that there would be an optimisation procedure which yields an ontology for an application processing semantic data. The second problem is that computer systems might not be suitable to describe human-oriented concepts. This problem has been tackled in the past by using fuzzy logic. One approach is described in [3], where a computational model which maps events and observations to an emotional state uses a fuzzy-logic representation. This paper is keeping the option of using an ontology based on fuzzy logic as one possible way of finding an ontology.

One important application of the proposed approach can be found in self-managed systems. The ‘executable reality’ approach as described in [4] is one example of a system which would benefit from the approach described in this article. ‘Executable reality’ is described in as “an extension of the (Mobile) Mixed Reality concept”. The described extension replaces part of the ‘static’ retrieval of information by computation of data using context sensitive business intelligence. When a device with such system is used at a location close by the sea concepts related to shores, harbours and seashells might become part of the active ontology. When the device is at a later time point used in a mountainous area the sea related concepts become partly redundant. If the device has a small storage capacity, the most optimal ontology will not contain these concepts any longer. A device which has, on the contrary, an abundant storage capacity but low processing power should keep the concepts stored to avoid computationally expensive changes in the ontology.

2 Optimisation

Optimisation problems are in general statements of problems where a best solution is to be found according to certain criteria and restrictions. The first paragraphs describe a few classes of global optimisation problems in order to introduce the Context-dependant Dynamic Multi-objective optimisation class in the last paragraph.

Static single-objective optimisation is considered a basic type of optimisation problems. The problem statement consists of a function which will be called f with domain D and range R. The domain of the function can be given explicitly as a set or described using constraints on a set. The set used for the range should have a total order relation $\leqslant$ defined on its elements, i.e. there is a transitive, antisymmetric, and total relation on the elements of R.

Definition 1. *An element $d \in D$ is optimal for a function f, i.e. $d \in opt(f) \Leftrightarrow \forall e \in D : f(e) \leqslant f(d)$*

The class of optimisation problems described in the previous definition can be extended to a multi-objective variant by allowing the range of the function to be a set of vectors of dimension n. The range of the function f is thus $R_1 \times R_2 \times \cdots \times R_n$ where we require a (strict) total order relation $<_i$ to be defined on each R_i. The domain of the function is a set D. Note that all single objective optimisation problems are multi-objective problems.

Definition 2. *An element $d \in D$ is multi-objective optimal for a function f, i.e. $d \in opt(f) \Leftrightarrow \forall e \in D \setminus \{d\} : (\exists i \in [1, n] : f(e)_i <_i f(d)_i)$*

Dynamic optimisation is essentially not different from solving a series of (multi-objective) static optimisation problems. The dynamism in the series is to be found in the way the function which is optimised or the constraints on the domain of the function being optimised are changing. The series can be represented by a function F, which maps the natural numbers to a set of pairs of a function and its domain. The functions in the tuples have a domain which is a subset of the total domain D the optimisation problem is working on. One way of solving the dynamic optimisation problem is by finding the optimal value for the functions for all possible values in $\mathbb{N}$ and then selecting the maximum value. We denote $opt'(F(t))$ to be the optimisation problem with function $F(t)_1$ and constraints $F(t)_2$.

Definition 3. *An element $d \in D$ is considered optimal for the dynamic optimisation of the function $F : \mathbb{N} \rightarrow (functions, constraints)$ if $\forall t \in \mathbb{N} : opt'(F(t)) \leq d$.*

Another possible definition follows:

Definition 4. *A function sol is the solution of the dynamic optimisation problem if $sol : \mathbb{N} \rightarrow D$ and $sol(t) = opt'(F(t))$*

In other words, a solution is such function which gives the optimal solution for each possible $t \in \mathbb{N}$.

Context-dependant dynamic multi-objective optimisation problems are an extension of the dynamic optimisation problems defined in the previous paragraph. In this case the domain of the function F is a series of what will be called 'contexts' instead of the natural numbers. The solution of the optimisation problem can be stated in a similar way as the definitions in the previous paragraphs. A solution of this type of problem indicates (multi-objective) optimal solutions in particular contexts, or analog optimal solutions over the range of all possible contexts.

3 Features of an Ontology

An application needs schemas or ontologies in order to give meaning to the semantic data it is processing. Different definitions of ontologies have been proposed. Gruber [5] defined, for instance, that "An ontology is an explicit specification of a conceptualization.", which allows for a very broad interpretation. A more concrete definition for description of ontologies is OWL2 endorsed by the World Wide Web Consortium. [6] There is a close correspondence (and sometimes even compatibility) between ontologies and description logic. OWL2 is for instance compatible with the description logic SROIQ. [7]

For this article, the concrete syntax of the ontology used is not of mayor importance. More important is the fact that an ontology has certain features. Examples of features of an ontology include coverage, cohesion and coupling. [8] In this article the properties which an ontology has independent of any context will be called the 'features' of the ontology. Some literature calls these properties 'quality' factors. The name quality will however be used in one of the following sections to describe the effect of features in a context. Other methods for measuring features of an ontology have been proposed by Burton-Jones et al. [9] and Yao et al. [10] and many other feature sets can be found in the literature.

4 Fuzzy Ontologies

For the representation of not exactly defined sets, fuzzy sets as described in "Fuzzy Sets" [11] can be used. The elements of a fuzzy set are members of the set according to a given membership function. This function defines for each element of the considered universe a grade of membership in the set. The grade of membership is a real value in the interval $[0, 1]$, where a value of 0 means that the element is not in the set and a value of 1 indicates that the element is in the set. Any value in between indicates up to which extend the element is a member of the set.

Calegari and Giucci used the concept of fuzzy sets to describe what they call 'Fuzzy Ontologies', 'Fuzzy Description Logics' and 'Fuzzy-OWL'. [12] This research showed it to be possible to describe ontologies which are not exactly known by using membership functions. Bobillo and Straccia [13] did a similar work, but used OWL 2, and proposed a concrete XML syntax for the extension.

5 Ontology Evolution

In research on databases it has been noticed that the schemas which are used change over time. Ontologies have similar properties. Changes in the domain, the conceptualisation and the specification are unavoidable and the ontology has to be changed accordingly. The domain changes because the real world changes, the conceptualisation because the perspective changes and the specification changes when an ontology is to be represented in a language with different semantics and expressiveness. The whole of these changes is called ontology evolution. Ontologies and database schemas are different concepts. Firstly, the ontology itself can also be part of the data and this way the data becomes self-descriptive. Secondly, ontologies are explicitly designed for reuse in other context as the initial context of creation and are, moreover, decentralised by nature. Lastly, ontology models have, in general, a richer set of properties available for describing the domain and quite often the border between the schema and instance data is blurry.[14]

The same article also describes concrete effects of ontology evolution on the data set. For instance removal of a class causes instances to have a less specific type, declaring classes disjoint makes instances that are in both classes invalid and defining a class as a subclass of another one adds new possible properties on the subclass.

Despite their differences, a more recent article by Hartung et al. [15] claims that ontology evolution has similar requirements as changes in database schemas evolution. Ontology evolution requires, for instance, "support for a rich set of changes, expressive mappings, update propagation to instances and dependant schemas/ontologies, versioning and user-friendly tools". The same article compares several approaches for managing and tracking ontology evolution in terms of the above mentioned criteria.

6 Context and Quality of an Ontology

In order to talk about the quality of an ontology, one needs to take the context within which it is used into account. This statement can be supported by an example. Imagine for instance that one would say that an ontology which contains more concepts, is better than one with less concepts. This could be true in certain situations. However, if one imagines a system which only uses the concepts which are mentioned in the ontology with less concepts, the smaller ontology might be even better because it uses less memory space.

One way of defining ontological quality is described in "Data Driven Ontology Evaluation" [16]. This method uses a combination of a corpus and the ontology to evaluate the quality of the ontology. The corpus is a textual description of the ontology, which form the basis of different approaches of measuring ontologies described. The paper further elaborates on the fact that there is more as one quality aspect with regard to ontologies. Factors like price to build, maintenance and re-use are highlighted as very influential. Furthermore, the article mentions that the quality might be subjective to time, location and other contextual factors.

A recent survey on ontology evaluation tools was performed by Aruna et al. [17] This survey puts a stress on more technical demands of an ontology in a working system. Technical properties surveyed are interoperability, turn around ability, performance, memory allocation, scalability, integration into frameworks and interfaces. The ontological properties are limited to language conformity and consistency.

Research on improving the quality of an ontology by transformation operations on an existing ontology is described in [18]. The idea is that certain quality criteria can be fulfilled better by a transformation of the existing ontology into a new, but equally valid ontology. Concrete, several transformations are described to improve the ontology in terms of homogeneity, totality of properties, stability over time, and explicitness (as opposed to inferred) and uniformity in properties. The size of the ontology and simplicity of queries is, according to the same article, only assessable in a context.

One way of measuring the quality of an ontology might be to compare the ontology with one or more sets of data which should be described by the ontology. This comparison can be done using either the open-world assumption which is typically made in the semantic web or a closed-world assumption.

Quality does not have to be a one dimensional property. The quality of an ontology in a context can thus be a multidimensional property. The more complete the context for optimisation is, the fewer dimensions the quality will have. A 'complete' context will lead to a one dimensional quality.

7 Quality of Ontology in a Context as a Dynamic Multi-objective Optimisation Problem

As argued in the previous section, it is only reasonable to make statements about the quality of an ontology in a given context. If for a certain context the quality for two ontologies is given, then it is possible to make a comparison between the two ontologies in that particular context. However, quality will not always be one-dimensional and hence analog to the multi-objective class of optimisation problems, it is not always possible to say that either the first or the second ontology is better.

When it is possible to compare the quality of ontologies in a context, then it is also possible to define what it means for an ontology to be optimal in a given context. Because of the fact that there is no total order on the quality of an ontology in a context, it will be necessary to consider the search for an optimal ontology for a given context as a multi-objective optimisation problem.

Let C be the set of contexts, O be the set of ontologies and Q be the set of qualities. Now we can define the optimisation as:

Definition 5. *The function sol is the solution of the context-dependant dynamic multi-objective problem of finding an optimal ontology for a given context* $\Leftrightarrow$ $sol : C \rightarrow O$ *and* $sol(c) = opt'(F(c))$

Where $F(C) \rightarrow (O \rightarrow Q)$ a function which maps the context c to a function which incorporates the context when evaluating the quality of an ontology and its associated domain.

It makes most sense to use the definition of context-dependant optimisation with a function since the system will be used in a real life setting and it is impossible to predict the optimal ontology over the whole life-time of the system at any point. It should also be noted that in any real system, the function *sol* can only be known partially and most likely by approximation. It is only known partially because the context is changing all the time and the future contexts are still unknown. Only an approximation can be found because no real system can react quick enough to all changes in a complex context in order to compute a new optimal ontology.

8 Future Research Directions

As said in the previous section, it is important to note that the context of the system in which the ontology is optimised will change dynamically over time. One important aspect of the context is the history of the system. The reason is that taking a new ontology into use causes a certain replacement overhead. A new ontology causing a big overhead has a lower quality in the context of the system. Cuenca Grau et al. [19] tried to reduce the cost of consistency checking of a new ontology by using the previously used ontology, i.e. the history of the system.

In this article we did not specify any concrete ontology notation to be used for this type of system. There is a need to evaluate the different possible classes of ontologies and see which properties of ontologies are affected in this type of optimisation. Depending on the type of ontology chosen, the system has different options for the development of its ontology. If the ontology would be for instance a fuzzy ontology, it might even be feasible to change only membership functions to adapt to changes in the context.

It is an open question how the system should search for an optimal ontology for a given context. Furthermore, this article refrained from defining a concrete definition of context and how it should be incorporated in the functions for optimisation. One popular choice for the incorporation of context is the use of weighted sum based methods. More research is needed to link particular features of an ontology to quality aspects, as well as how the context influences this.

Next, it seems reasonable to look whether it is possible to notice trends in the evolution of the ontology. A system could then take the trends into account when assessing a new ontology or predict resource consumption in the future.

Lastly, it is interesting to consider what would happen when two separate systems have their isolated evolutions of the ontology. Questions in that kind of situation include what should be done to align these ontologies, what to do with discrepancy between the ontologies, whether these systems can interact if they are using different ontologies, etc...

9 Conclusion

The aim of this article was to show initial findings to solve the problems of interoperability in the Semantic web in case ontologies are not shared among applications and how to allow these applications to work with more 'humanised' concepts. The first problem was reduced to a formulation of the finding of an optimal ontology in a given context in section 7. This context includes for instance the data processed by the application, the past used ontologies and constraints related to the system. The challenge of allowing humanised concepts is attempted by allowing fuzzy logic to be used in this kind of system.

This article was financially supported by the 'Cloud Software Program' of TiViT Oy. We would like to thank 'Intelligent Precision Solutions and Services Oy' and in particular Sami Helin for proposing the initial 'Cloud Communication Channel' business case in which this research idea was elaborated.

References

1. Carroll, J.J., Klyne, G.: Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C (February 2004) http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.
2. Marshall, C., Shipman, F.: Which semantic web? In: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, ACM (2003) 57–66
3. El-Nasr, M.S., Yen, J., Ioerger, T.R.: Flame—fuzzy logic adaptive model of emotions. Autonomous Agents and Multi-Agent Systems **3** (2000) 219–257 10.1023/A:1010030809960.
4. Terziyan, V., Kaykova, O.: Towards" executable reality": Business intelligence on top of linked data. In: BUSTECH 2011, The First International Conference on Business Intelligence and Technology. (2011) 26–33
5. Gruber, T., et al.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2) (1993) 199–220
6. Group, W.O.W.: OWL 2 web ontology language document overview. Technical report, W3C (October 2009) http://www.w3.org/TR/2009/REC-owl2-overview-20091027/.
7. Patel-Schneider, P.F., Motik, B., Grau, B.C.: OWL 2 web ontology language direct semantics. W3C recommendation, W3C (October 2009) http://www.w3.org/TR/2009/REC-owl2-direct-semantics-20091027/.
8. Ouyang, L., Zou, B., Qu, M., Zhang, C.: A method of ontology evaluation based on coverage, cohesion and coupling. In: FSKD. (2011) 2451–2455
9. Burton-Jones, A., Storey, V.C., Sugumaran, V., Ahluwalia, P.: A semiotic metrics suite for assessing the quality of ontologies. Data Knowl. Eng. **55**(1) (October 2005) 84–102
10. Yao, H., Orme, A.M., Etzkorn, L.: Cohesion Metrics for Ontology Design and Application. Journal of Computer Science **1**(1) (2005) 107–113
11. Zadeh, L.: Fuzzy sets. Information Control **8** (1965) 338–353
12. Calegari, S., Ciucci, D.: Fuzzy ontology, fuzzy description logics and fuzzy-owl. In: Proceedings of the 7th international workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory. WILF '07, Berlin, Heidelberg, Springer-Verlag (2007) 118–126
13. Bobillo, F., Straccia, U.: Fuzzy ontology representation using owl 2. CoRR **abs/1009.3391** (2010)
14. Noy, N., Klein, M.: Ontology evolution: Not the same as schema evolution. Knowledge and Information Systems **6**(4) (2004) 428–440
15. Hartung, M., Terwilliger, J., Rahm, E.: Recent advances in schema and ontology evolution. Schema Matching and Mapping (2011) 149–190
16. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data driven ontology evaluation. In: International Conference on Language Resources and Evaluation (LREC 2004). (2004)
17. Aruna, T., Saranya, K., Bhandari, C.: A survey on ontology evaluation tools. In: Process Automation, Control and Computing (PACC), 2011 International Conference on. (july 2011) 1 –5
18. Mostowfi, F., Fotouhi, F.: Improving quality of ontology: An ontology transformation approach. In: Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on, IEEE (2006) 61–61
19. Cuenca Grau, B., Halaschek-Wiener, C., Kazakov, Y.: History matters: Incremental ontology reasoning using modules. The Semantic Web (2007) 183–196