Ontology Verification Using Contexts

Aviv Segev and **Avigdor Gal**
Technion – Israel Institute of Technology
{asegev@tx, avigal@ie}.technion.ac.il

Abstract. Ontologies have become the de-facto modeling tool of choice, used in a variety of applications and prominently in the Semantic Web. Their design and maintenance, nevertheless, have been and still are a daunting task. As a result, ontologies quickly become underspecified. Therefore, if ontologies do not evolve, the semantic infrastructure of the information system can no longer support the changing needs of the organization. In this work we provide a model to semi-automatically support relationship evolution in an ontology using contexts. We propose to use (machine-generated) contexts as a mechanism for quantifying relationships among concepts. To do so we compare the contexts that are associated with the ontology constructs. On a conceptual level, we introduce an *ontology verification* model, a quantified model for automatically assessing the validity of relationships in an ontology. We motivate our work with examples from the field of eGovernment applications. To support our model with an empirical analysis, we provide a mapping of an ontology operator for defining relationships into context relationships, using real-world traces of RSS.

1 INTRODUCTION

Ontologies have become the de-facto modeling tool of choice, used in a variety of applications and prominently in Semantic Web applications. For example, ontologies can be used in discovering Web services [10]. Ontology design, nevertheless, has been and still is a daunting task. It requires collaboration of domain experts with ontology engineers, which may consume many organizational resources in terms of both time and monetary units. Once the ontology is designed, evolving it becomes difficult due to the need for availability of domain experts on the one hand, and costs related with hiring ontology engineers on the other hand. To illustrate this point, consider an eGovernment application, for which an ontology was designed and tailored by an ontology engineer. Once the ontology is installed, changes in the real world require a renewed collaboration of civil servants with ontology engineers to reflect such changes in the ontology. A typical outcome of such difficulties is that ontologies quickly become underspecified. New concepts are introduced in the domain while others become obsolete. Also, shifts of focus in the application domain require the refinement of a concept into a hierarchy of concepts, while in other cases hierarchies should be collapsed. Meeting these challenges requires ontologies to evolve or else the semantic infrastructure of the information system can no longer support the changing needs of the organization.

In [9] we introduced a model for compensating for ontology underspecification using a combination of ontologies with contexts. *Contexts* were defined to be first class objects [5] and will be formally presented later in this work. As an example, a context can be defined to be a set of words, possibly associated with weights that represent the relevance of a word to a document. Ontologies and contexts are both used to model different perspectives of a domain (views). Ontologies represent shared models of a domain and contexts are local views of a domain. We also promote an orthogonal classification in which ontologies are considered a result of a manual effort of modeling a domain, while contexts are system generated models [8]. Ontologies and contexts are joined together, as formally described in [9]. In a nutshell, each concept in an ontology is represented by a name and a context. In this model, contexts serve as an easy-to-use "semantic glue," in which underspecifications are compensated for with a syntactic, machine generated context, which highlights the intentions of a local designer when using a specific ontology concept, possibly differently from the way it is semantically captured in the ontology using relationships.

In this work we provide a model and an example of an algorithm to semi-automatically support relationship evolution in an ontology using contexts. The main motivation for this work stems from the difficulty in supporting ontology evolution. Specifically, this problem was raised within the framework of TerreGov, a European eGovernment project. In this project, ontologies serve as the driving force behind the application and thus affect government processes and Web services, among other things. Therefore, we propose to use (machine-generated) contexts as a mechanism for quantifying relationships among concepts. Specifically, given an ontology operator (*e.g.*, *link subclass*, representing the knowledge that an instance of one concept is included in an instance of another) and operands (*e.g.*, two concepts or classes), we aim at quantifying the extent to which this relationship is valid. We do so by comparing the contexts that are associated with the operands. We believe that such a solution would significantly assist in the support of ontology design and evolution.

The main contribution of this work is thus twofold. On a conceptual level, we introduce an *ontology verification* model, a quantified model for automatically assessing the validity of relationships in an ontology. On an algorithmic level, we provide an example of a mapping of ontology operator for defining relationships into context relationships. We motivate our work with examples from the eGovernment domain. However, due to the absence of large scale data sets for this domain, we support our model with an empirical analysis using real-world traces of RSS data.

The rest of the paper is organized as follows. We start with preliminaries, formally defining ontologies and contexts in Section 2. In Section 3 we introduce the ontology verification model, followed by an example of a mapping of the ontology verification problem to contexts in Section 4. We conclude with related work in Section 5 and a short summary in Section 6.

2 ONTOLOGIES AND CONTEXTS

Banerjee [1] defined a *root class* as an object that represents anything from a simple number to a complex entity. An edge between a node and a child node in a class represents an IS-A relationship. Objects that belong to a class are called *instances* of that class. A class describes the *form* (instance variables) of its instances and the *operations* (methods) applicable to its instances.

According to Gruber [2], an *ontology* is an explicit specification of a domain conceptualization. Several models for ontologies exist; we follow here that presented in [2]. In the discussion below, we assume reader familiarity with basic concepts in conceptual modeling.

We define a *context* $\mathcal{C} = \left\{\{\langle c_{ij}, w_{ij}\rangle\}_j\right\}_i$ as a set of finite sets of descriptors c_{ij} from a domain $\mathcal{D}$ with appropriate weights w_{ij}, representing the importance of c_{ij}. For example, a context $\mathcal{C}$ may be a set of words (hence, $\mathcal{D}$ is a set of all possible character combinations) defining a document Doc, and the weights could represent the relevance of a descriptor to Doc. In classic Information Retrieval, $\langle c_{ij}, w_{ij}\rangle$ may represent the fact that the word c_{ij} is repeated w_{ij} times in Doc.

The context of a class is defined as a set of contexts describing instances that belong to this class. Documents are not instances but represent them. Following [9], we define a class context $\mathcal{C}_{CL}$ of a class CL to be the union of its instance contexts.

Segev and Gal [9] aimed at formalizing the inter-relationships between an ontology, a manually generated domain model, and contexts, partial and automatically generated local views. According to their work, a context can belong to multiple context sets, which in turn can converge to different ontology concepts. Thus, one context can belong to several ontology concepts simultaneously. The appropriate interpretation of a context leads to its relevance to different given concepts.

3 ONTOLOGY VERIFICATION USING CONTEXTS

Ontology verification is the process by which semantic relationships are identified. We term this process verification, since we assume an ontology exists and may need to evolve. Therefore, semantic relationships in an ontology need to be continuously monitored and if necessary, revised. Here we follow the work of [6] on ontology changes and assume a given closed set of operators OT, to be applied on a set of operands OD, taken from the set of all ontology elements. As an example, a change operator may be the *disjoint* operator, resulting in the creation of a semantic relationship called "disjoint" between two classes, given to it as operands.

Figure 1 provides a pictorial representation of the process. Formally, ontology verification is a function $OV : OT \times OD^* \rightarrow [0, 1]$. Ontology verification is given as input a hypothesis regarding the possible operator to be applied to one or more operands and returns a level of certainty μ regarding the truth in this hypothesis. A certainty of 1 indicates full certainty in the hypothesis, while a certainty of 0 means that the hypothesis is definitely incorrect. In Figure 1, the ontology verification function determines that the disjointedness of classes CL_1 and CL_2 has a certainty level of 0.9. An example of the use of the model can be a user who would like to analyze a local government concept relationship. The user could supply a set of documents representing two concepts and could receive a verification level based on this representative set of documents.

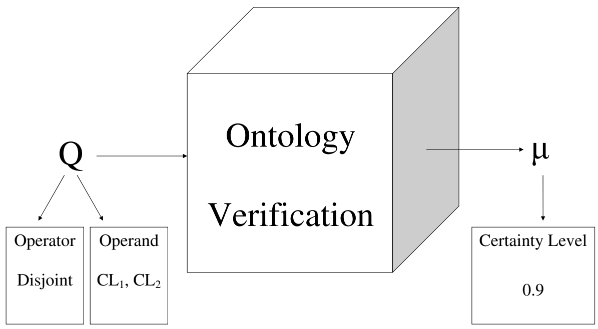

Figure 1. Ontology Verification Model

4 EXPERIENCES WITH CONTEXT BASED ONTOLOGY VERIFICATION

Having introduced ontology verification, we now focus on the details of change operators. Noy and Klein [6] describe a set of 22 ontology change operators and their impact on ontology elements (both classes and instances) using ontology relations defined in [2]. We take one of their ontology change operators and use it as an example.

Our experiences are based on data from the RSS news data trace. In this data trace, data were originally partitioned to topics with no ontological relationships. The RSS trace was collected during August 2005 from the CNN Web site. We chose 10 news topic categories for the data, when each RSS news header or news descriptor constitutes a datum. We generated a context for each datum and each class using an automatic context extraction algorithm [8]. The number of context descriptors generated from each datum was set to 10. The data size used for RSS varied from 73 to 1,911 per class.

In our experiment we calculated for each class the number of contexts that overlapped with the other nine classes. This asymmetric comparison gave us for any two classes CL_i and CL_j the metric of $\left|\mathcal{C}_{CL_i} \cap \mathcal{C}_{CL_j}\right|$ and $\left|\mathcal{C}_{CL_i} \cup \mathcal{C}_{CL_j}\right|$.

Given two classes, CL_i and CL_j, if CL_i is a subclass of CL_j, then its context should be contained in the context of CL_j. This is because an instance of CL_i is also an instance of CL_j and therefore has a broader context than an instance of the superclass. Therefore, we compute the certainty of a hypothesis that CL_i is a subclass of CL_j to be

$$\mu_{\textit{Sub-Sup}} = \frac{\left|\mathcal{C}_{CL_i} \cap \mathcal{C}_{CL_j}\right|}{\left|\mathcal{C}_{CL_j}\right|}$$

Our experience involves an analysis of hierarchy linking. Figure 2 presents the RSS class relations hierarchy created for $\mu_{\textit{Sub-Sup}} \geq 0.8$ and $\mu_{\textit{Sub-Sup}} \geq 0.5$. As the value of $\mu_{\textit{Sub-Sup}}$ decreases, the hierarchy and the relations between the classes become more elaborated. For example, in the RSS data for $\mu_{\textit{Sub-Sup}} \geq 0.8$ the superclass Money Latest has four subclasses. If we examine the same classes for a lower verification level of $\mu_{\textit{Sub-Sup}} \geq 0.5$ we receive a three level hierarchy.

Table 1 compares the certainty level of the *Superclass-Subclasss* operator, for two class pairs in the RSS data set. When evaluating the classes Money Latest and Money News International, there is a high $\mu_{\textit{Sub-Sup}}$ level.

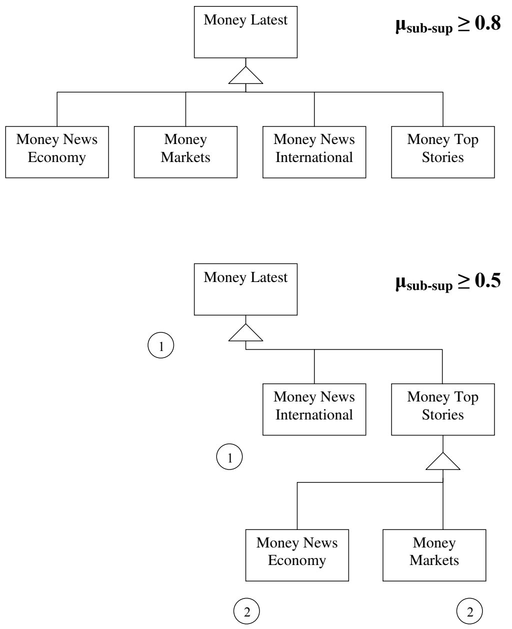

Figure 2. RSS Relations

5 RELATED WORK

A formal mathematical framework that delineates the relationships between contexts and ontologies is presented in [9]. To deal with the uncertainty associated with automatic context extraction from existing instances, such as documents, a ranking model was provided, which ranks ontology concepts according to their suitability with a given context.

A semi-automated method for ontology evolution using documents clustering was proposed in [11]. From the results of the clustering ontology enrichments and updates are extracted. In contrast to the above work, which is based on a single word ontology concept description, we use a set of contexts describing each ontology class.

Noy and Klein [6] defined a set of ontology-change operations and their effects on instance data used during the ontology evolution process. They describe ontologies schemas and database schemas from the point of view of evolution and highlight the main differences between them. We presented a model that shows how these ontology change operations can be verified based on context.

Tools for merging and aligning ontologies, such as SENSUS [3], PROMPT [7], and Cyc [4], have been developed in the past. These tools generally present a set of basic operations that are performed during the mergence and alignment of ontologies and that determine the effects that each of these operations has on the process.

A work on multi-contextual ontology evolution [12] defines a set of properties that by semantic autonomy must hold at the same time.

6 CONCLUSION

This work presents a model and a set of algorithms to semi-automatically support ontology relationship evolution using contexts.

Class Sets	Link Subclass
Money Latest	86.7%
Money News International	19.8%
Money News Economy	19.5%
Money Markets	24.3%

Table 1. Operator μ Verification RSS

Given an ontology operator and operands, the model provides the quantification of the extent to which the relationship is valid. The model is supported by empirical analysis, using initial experiences with real-world RSS traces. The experiences with these traces show how relationships between the classes can be created and modified. Preliminary empirical results show that our model can provide good estimations of the need for ontology changes.

To recap, the main contribution of this work is both conceptual and algorithmic. We present an ontology verification model, a quantified model for automatically assessing the validity of relationships in an ontology, and we also provide a mapping of several ontology operators for determining relationships among classes.

The results of this work will be embedded as part of the TerreGov solution. Future research will examine the model performance on eGovernment data and other large data sets. In addition, we plan on extending the model to include additional operators.

REFERENCES

[1] J. Banerjee, H.-T. Chou, J. Garza, W. Kim, D. Woelk, and N. Ballou, ‘Data model issues for object-oriented applications’, *ACM Transactions on Office Information Systems*, **5**(1), 3–26, (1987).

[2] T. R. Gruber, ‘A translation approach to portable ontologies’, *Knowledge Acquisition*, **5**(2), (1993).

[3] K. Knight and S. K. Luk, ‘Building a large-scale knowledge base for machine translation’, in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, (1994).

[4] D. B. Lenat, ‘Cyc: A large-scale investment in knowledge infrastructure’, *Communications of ACM*, **38**(11), 33–38, (1995).

[5] J. McCarthy, ‘Notes on formalizing context’, *In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, (1993).

[6] N. F. Noy and M. Klein, ‘Ontology evolution: Not the same as schema evolution’, *Knowledge and Information Systems*, **6**(4), 428–440, (2004).

[7] N. F. Noy and M. A. Musen, ‘The prompt suite: Interactive tools for ontology merging and mapping’, *International Journal of Human-Computer Studies*, **59**(6), 983–1024, (2003).

[8] A. Segev, ‘Identifying the multiple contexts of a situation’, in *Proceedings of IJCAI-Workshop Modeling and Retrieval of Context (MRC2005)*, (2005).

[9] A. Segev and A. Gal, ‘Putting things in context: A topological approach to mapping contexts and ontologies’, in *Proceedings of AAAI-Workshop Workshop on Contexts and Ontologies: Theory, Practice and Applications*, (2005).

[10] E. Toch, A. Gal, and D. Dori, ‘Automatically grounding semantically-enriched conceptual models to concrete web services’, in *ER*, eds., L.M.L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, volume 3716 of *Lecture Notes in Computer Science*, pp. 304–319. Springer, (2005).

[11] G. Tsatsaronis, R. Pitkanen, and M. Vazirgiannis, ‘Clustering for ontology evolution’, in *Proceedings of the 29th Annual Conference of the German Classification Society (GfKl 2005)*, (2005).

[12] M. Zurawski, ‘Reasoning about multi-contextual ontology evolution’, in *Proceedings of the First International Workshop on Context and Ontologies: Theories, Practice and Applications, The Twentieth National Conference on Artificial Intelligence (AAAI-05)*, (2005).