A View of OWL from the Field: Use cases and Experiences

Aaron Kershenbaum[1], Achille Fokoue[1], Chintan Patel[1,3], Christopher Welty[1], Edith Schonberg[1], James Cimino[3], Li Ma[2], Kavitha Srinivas[1], Robert Schloss[1], and J William Murdock[1]

IBM T. J. Watson Research Lab[1], Hawthorne, USA
{aaronk, achille, copatel, welty, ediths, ksrinivs, rschloss, murdockj}@us.ibm.com
IBM China Research Lab[2], Beijing, China
malli@cn.ibm.com
Department of Biomedical Informatics[3], Columbia University, New York, USA
{cop2101, jjc7}@columbia.edu

Abstract. In this paper, we describe our experiences with Semantic Web applications from the domain of life sciences, text mining and software engineering. In these domains, the state-of-the-art is limited to the use of simple taxonomies. This is partly because a sufficient set of use cases has not yet been developed to demonstrate the value of using more expressive languages (such as OWL) to add value in these domains. We are starting to catalog a set of such use cases, and we describe three concrete use cases in this paper.

1. Introduction

The Web Ontology Language (OWL) provides rich constructs to represent a variety of constraints in a given domain. A large number of real-world ontologies, with very few exceptions, tend towards minimal use of these expressive constructs. The Gene Ontology, SNOMED-CT, and the NCI-Thesaurus, for example, are large ontologies that do little more than provide a taxonomy. FOAF, the most widely used ontology on the semantic web, does little more than define the *Person* class and the "friend" relation.
There is a real cost to making more use of the expressive power of OWL, such as development/maintenance, training in logic, scalability, etc. What is also unclear to practitioners in the field is the benefits of such increased expressivity. We have started to catalog a set of use cases for the use of OWL ontologies. In this paper, we present three concrete real-world use cases, and discuss the cost-benefit tradeoffs involved in increasing the expressiveness of the ontology.

2. Applications

In the spirit of showing the value added by migrating taxonomies to OWL in the real world, we first describe the problem domain and present the value added by the addition of OWL constructs. Specifically, we were able to identify modeling errors in a medical ontology, represent clinical trials as Abox queries and eliminate the noise in text-analysis of unstructured data using expressive features of OWL.

2.1. Discovery of Modeling Errors in Real-world taxonomies

Medical Entities Dictionary (MED) [2] is a terminology that is used at the Columbia Presbyterian Medical Center, for the querying of patient records for various disease states/conditions. MED uses frame-based logic and contains 100,210 concepts and 261 slots. We started by transforming the laboratory concepts present in MED into OWL. Specifically, we modeled just the concepts related to laboratory tests in MED as follows: a laboratory test can be fully specified by the biochemical substance it measures and the sample that is being assessed in the test. We converted corresponding MED slots - 1) entity-measured and 2) assesses-sample into definitional properties (i.e. necessary and sufficient) for 10,981 lab concepts. Next, we used DL subsumption reasoning to classify MED. In comparing the classified hierarchy for such defined concepts with the original taxonomy in MED, we found 44 additional subsumptions for laboratory concepts. On manual analysis of newly inferred subsumptions, 26 were correct subsumptions i.e. the concepts actually had a subclass relationship, as confirmed by a domain expert. Interestingly, the false positives revealed systematic modeling errors.
The important result here is not that we identified these modeling errors due to the increased expressivity of DL. More important is our finding that the missed subsumptions could have cost the hospital many missing results in various decision support and infection control systems that routinely use MED to screen patients.

2.2. Selection for Clinical Trials and Tracking Infectious Diseases

Clinical trials are systematic studies in human patients aimed at determining the safety and effectiveness of new or unproven therapies. Low participation rate is a common problem in conducting clinical trials. For example, less than 5% of eligible patients participate in most cancer trials and less than 10% in many cardiovascular trials [1]. Given electronic medical records, we could potentially “match” the criteria in trials to records in clinical databases and alert the physician or patients about available trial options. A second common real-world need is the tracking of patients with infectious diseases (e.g., SARs).

2.3. OWL for the selection of patients

In the recently defined WHO standard[3] for clinical trials, all registrants are required to provide key inclusion and exclusion criteria for selecting eligible patients. *ClinicalTrials.gov* explicitly mentions the criteria for all their trials. What is interesting about these trials is that they usually specify certain exclusion criteria, which can only be expressed by the negation of complex concepts, as shown by a simplified example for *Disorder of Prosthetic Cardiac Valve* below (Figure 1). Note that these definitions of disease map to SNOMED-CT, which is now a standard ontology for the healthcare domain. The interesting point about this use case is that although SNOMED-CT is itself OWL-Lite, realistic use cases of SNOMED-CT require the use of negation.

An example from clinicaltrials.gov - NCT00286182

Inclusion Criteria:
Heart failure and a Normal Ejection Fraction (HFNEF)
...
Exclusion Criteria:
Disorder of prosthetic heart valve …

All exclusion criteria become negated concepts and create a complex expression

DL-membership query using SNOMED-CT

Heart failure ⊓ *(Cardiac ejection fraction* ⊓ *∃hasInterpretation.Normal)*
⊓ ¬ Disorder of prosthetic cardiac valve

where ¬Disorder of prosthetic cardiac value ≡ ¬(Heart Valve Disorder ⊓ ∃after.(∃directDevice. Prosthetic Implant ⊓ ∃method. Insertion) ⊓ ∃ associatedWith. Surgical Proc)

Fig. 1. Representation of clinical trials criteria as an ABox query

2.4. Multi-Annotator Text Analytics

The Unstructured Information Management Architecture (UIMA)[4] is one of the widely used tools for text-mining applications. UIMA provides an extensible framework that allows integration of different entity and relationship annotators. The annotators extract various entity and relationship phrases from free text e.g. *<John:Man>* *<employeeOf>* *<IBM:Company>*. The entity and relationship types are mapped to an OWL ontology (NIMD) that provides semantic underpinnings to these types e.g. *Company* β *Organization*.

Co-reference and Noisy Output

UIMA incorporates a co-reference resolution component that is responsible for identifying instances that refer to same individual within or across the text documents e.g. in the sentence: "*John walked on the beach; he likes sun*", <*John*> and <*he*> actually refer to the same person. Sometimes, the co-reference component might incorrectly merge two different instances to the same individual (Bush as shown in Figure 2).

Annotator1: Instance_1:Person	George W. Bush (1) is the current president of United States (2). In one of his visits to Houston, he landed at George Bush airport (3)…
Annotator2: Instance_1 <citizenOf> USA:Country	
Annotator3: Instance_1 <basedIn> Houston:GeoPolitical Entity	

domain(citizenOf) = Person

domain(basedIn) = Facility ⊔ Organization

Person ⊑ ¬Facility ⊓ ¬Organization

Fig. 2. Clean up of co-reference errors using ontology constraints

The set of constraints defined in the NIMD ontology allows clean up of such incorrect mergers. In our experiments[5] with the text-analysis data consisting of 1,999,787 triples, we found 34,490 triples that led to the inference that individuals were *Persons* and *Organizations* at the same time. This type of inconsistencies can be corrected by eliminating triples that result in an inconsistency, even as the triples are generated by the annotators, and added incrementally to the knowledge base.

2.5. Semantic Traceability

The Semantic Traceability project aims at providing an enterprise-wide understanding of relationships between various assets (artifacts, resources, people/roles). The main requirement was to provide extensible and efficient mechanisms to keep track of relationships between all information assets of an enterprise for the purpose of understanding dependencies, querying and performing impact analyses, assuring compliance, reducing unneeded duplication, and for n-stage model-driven design.

In the first phase of the project, an inexpressive OWL Lite TBox was defined to capture the semantics of important relationships between various software assets, and a large ABox was generated from a program analysis tool that extracted relationships between software artifacts. By increasing the expressiveness of the TBox (e.g. by defining *ConcreteImplementation* as the complement of an *AbstractImplementation*), an inconsistency in the ABox was detected. This problem was successfully traced back to the internal model of the analysis tool which, due to its naming scheme of nested java classes, could not distinguish two nested classes with the same non-qualified name defined in two different classes of the same package.

3. Conclusion

We have started cataloging a set of use cases that demonstrate the use of OWL in domains where reasoning is currently minimally used (as taxonomies). This paper describes a small set of real-world applications that can benefit from a migration to OWL.

References

1. Simes RJ. Clinical trials and "real-world" medicine. Trial evidence best informs real-world medicine when it is relevant to the clinical problem. Med J Aust. 2002 Oct 21;177(8):410-1.
2. Cimino JJ. From data to knowledge through concept-oriented terminologies: experience with the Medical Entities Dictionary. J Am Med Inform Assoc. 2000 May-Jun;7(3):288-97.
3. WHO International Clinical Trials Registry Platfom, http://www.who.int/ictrp/en/ (accessed July 27, 2006)
4. Unstructured Information Management Architecture, http://www.research.ibm.com/UIMA/ (accessed July 27, 2006)
5. Fokoue, A., Kershenbaum A., Ma, L. SHIN ABox Reduction, In Proc. of the 2006 Description Logic Workshop (DL 2006). CEUR (http://ceur-ws.org/), 2006.