OWL Support for (Some) Non-Deductive Scenarios of Ontology Usage

Vojtěch Svátek[1], Miroslav Vacura[1], Martin Ralbovský[1], Ondřej Šváb-Zamazal[1], and Bijan Parsia[2]

[1] University of Economics, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic
{svatek|vacuram|ralbovsm|ondrej.zamazal}@vse.cz
[2] University of Manchester, Oxford Road, Manchester M13 9PL, UK
bparsia@cs.man.ac.uk

Abstract. Applications of ontologies exist that go beyond standard deductive reasoning and rather have the character of empirical discovery in knowledge/data. We analyse the inventory of OWL with respect to two such applications, namely to pattern-based ontology matching and to ontology-aware knowledge discovery from databases.

1 Introduction

Most design decisions about the OWL standard are motivated by its DL semantics, reflecting deductive (in the general sense) scenarios of ontology exploitation. We however assume that OWL ontologies will be frequently used and sometimes even built by people primarily interested in computerised processing of information beyond the scenarios foreseen by logicians. The inventory of OWL could possibly be revisited with such applications in mind. We elaborate on two such scenarios we exploited in our research: *pattern-based ontology matching* and *ontology-based knowledge discovery from databases* (KDD). Other areas worth similar analysis could be e.g. ontology retrieval over the web [2] or ontology-based information extraction [3, 6].

2 Ontology Matching

By our experience from the Ontology Alignment Evaluation Initiative (OAEI),[3] ontology matching systems often fail on discrepancies that are not due to inherently different conceptualisations but due to different logical patterns (including those listed in `http://www.w3.org/2001/sw/BestPractices/OEP/`) used when adapting the conceptualisations to the restricted language. Let us present two such cases: indicating *reified n-ary relations* and capturing the *criteria of concept partition*. To illustrate them, we borrow from the 'conference organisation' domain as one of those examined within OAEI.[4]

[3] `http://oaei.ontologymatching.org/2007/`
[4] `http://nb.vse.cz/~svabo/oaei2008/`

Problem A.1 A reviewer can submit a review of a paper. There is inherently a ternary relationship between the reviewer, review and paper. The modeller of ontology A may use the SWBPD n-ary relation pattern:[5] creating a concept such as `ReviewSubmission`, with properties `reviewSubmitted`, `submittedBy` and `reviewForPaper`. The modeller of ontology B may however prefer a simpler (though lossy) solution: creating multiple properties expressing alternative variants of review, e.g. `submittedPositiveReview` and `submittedNegativeReview`. If the `ReviewSubmission` concept were annotated as 'reified relation' in ontology A, the task for a (pattern-aware) mapping system, aligning A with B via a heterogeneous mapping [4], and possibly enriching one other via a re-engineering pattern, would be easier.

Problem A.2 Submitted papers can be under review, accepted or rejected. At the same time, papers can be submitted as e.g. full papers, position papers or posters. In ontology A, this can be done by class `SubmittedPaper` having disjoint partition subclasses such as `PaperUnderReview`, `AcceptedPaper` and `RejectedPaper`, as well as `FullPaper`, `PositionPaper` and `Poster`. In ontology B, the same can be done using two data properties for the `SubmittedPaper` class: `phase` and `category` (with values analogous to the classes above). If the two disjoint partition axioms in ontology A were annotated as (informally) `criterion=phase` and `criterion=category`, respectively, the task for a mapping system aligning A with B would be easier, again yielding heterogeneous mappings (between an axiom and a data property).

3 Ontology-Based KDD

In the context of KDD, formal ontologies allow, among other, to form the data mining task more easily and accurately and to constrain the search space [7]. In the context of data pre-processing for *association mining*,[6] as the most critically needed part of domain information were identified *important values dividing the domain* of a data attribute, and *attribute groupings*. We illustrate them on medical examples inspired by [8].

Problem B.1 For the human blood pressure, values that separate normal blood pressure from hypertension and hypotension could be used to create categories of the respective attributes; such categories could have solid chance for yielding strong associations and also make these hypotheses more easily interpretable by experts. Cutting values can be implicitly derived from restrictions on data properties. For example, if class `Hypertension_Observation` is defined as having the value of data property `hasSystolicBPValue` higher than 140 and the value of data property `hasDiastolicBPValue` higher than 90 then these values can be suggested as cutting values in the data preparation phase of KDD. The designer of the ontology should however also be able to specify cut values of a data property directly.

[5] `http://www.w3.org/TR/swbp-n-aryRelations`

[6] We do not see the this information specifically relevant to association mining; it can probably be used in numerous other KDD tasks.

Problem B.2 Attributes such as `Angina_Pectoris`, `Myocardial_Ischemia` and `Hypertension` should be grouped together, as they all correspond to cardiovascular diseases. The same holds for attributes such as `Height`, `Weight` or `Girth` (biometric measures). While the former are likely to appear as classes in an OWL ontology, the latter could sensibly be data properties.

4 Possible Solutions

It seems that (entities or axioms in) OWL ontologies could be enriched with the required information (expressing e.g. relation reification, partition criteria, property cut values and property groupings) in at least four ways:

Solution S0 Enriching the specification of the OWL language itself
Solution S1 Using taxonomic inheritance
Solution S2 Using meta-modelling
Solution S3 Using the forthcoming annotation system of OWL [1].

The first option is clearly hard to put through due to language parsimony concerns. We will thus not consider it for the moment.

The second option is essentially harmless: additional information is attached to the given entity by subordinating it to an entity from a specific 'structural' ontology. This however only allows to assign binary features (and not e.g. numerical values) and only to atomic entities (not to axioms).

The third option consists in introducing new properties that have as their domain either OWL *meta-classes* such as `rdfs:Class` or `owl:DataProperty` or their purposefully defined subclasses. This allows to enrich all entities that are instances of the given meta-class; it however immediately lifts the ontology to OWL Full and also contaminates the domain model with information irrelevant to deduction. A prototype of such practice is the way of modelling so-called meta-properties in *OntoClean* [5]: as discussed in [1], the OntoClean methodology requires putting properties on classes in such a way that OntoClean conflicts manifest as an inconsistency. However, it seems likely that modelers need to keep OntoClean conflicts and domain modeling inconsistencies separate. Notably, an OntoClean conflict could be due to an incorrect subsumption; making the ontology inconsistent makes it harder to understand and debug that subsumption.

The fourth option looks most universal; it however depends on the final structure of the OWL 2 annotation system. An advantage might be the possibility to store additional information in a *separate file* (information space).

Let us now relate the four problems above to the solutions S1-S3:

A.1 Mere concepts have to be annotated. It is thus possible to use S1, namely, to create a special ontology with class such as `ReifiedRelation`, to import this ontology, and to subordinate the `ReviewSubmission` class to it. A disadvantage of this solution is the fact that an ontological concept was introduced for a notion that is merely anchored in the language (here, OWL as language lacking n-ary relation constructs) and has no real-world semantics. Alternatively, S3 could presumably be used.

A.2 Disjoint partition axioms have to be annotated. The only applicable solution seems to be S3.

B.1 In our implemented prototype [8] we used S2, i.e. meta-modelling: data properties such as `domainDividingValue` were assigned to `rdfs:Class`.[7] Alternatively, S3 could presumably be used.

B.2 As OWL allows to specify a taxonomy for both classes and properties (including data properties), no information external to the actual domain ontology is presumably needed in most cases, as attribute grouping can reflect the taxonomic closeness of the respective ontology entities. If necessary, an ontology specifically expressing 'groupings relevant to data mining' could be imported, with artificial classes such as `Group_Concept`, i.e. realizing S1. If such 'hard-coding' were undesirable, S2 could alternatively be used, via introducing an object property such as `sameAttributeGroup` having metaclasses in its domain and range. Finally, S3 could be applied in a similar manner as S1.

In general, it seems that exploiting a *rich annotation system* would help avoid possible negative impact on formal complexity and/or domain accuracy.

This work was partially supported by the IGA VSE grant no.20/08 "Evaluation and matching ontologies via patterns" and by the CSF project no.201/08/0802, "Application of Knowledge Engineering Methods in Knowledge Discovery from Databases".

References

1. Annotation System. OWL WG Work-in-Progress document, `http://www.w3.org/2007/OWL/wiki/Annotation_System`.
2. d'Aquin, M., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E.: Characterizing Knowledge on the Semantic Web with Watson. In: EON'07 Workshop at ISWC'07, Busan, Korea, 2007.
3. Embley, D. W., Tao, C., Liddle, D. W.: Automatically extracting ontologically specified data from HTML tables of unknown structure. In: Proc. ER '02, pp. 322–337, London, UK, 2002.
4. Ghidini, C., Serafini, L.: Reconciling concepts and relations in heterogeneous ontologies. In: Proc. ESWC 2006, Budva, Montenegro, LNCS 4011, 50–64.
5. Guarino, N., Welty, C.: An Overview of OntoClean. In: Staab, S., Studer, R., eds.: The Handbook on Ontologies, Springer-Verlag, 151-172.
6. Labský, M., Nekvasil, M., Svátek, V., Rak, D.: The Ex Project: Web Information Extraction using Extraction Ontologies. In: Proc. PriCKL'07, ECML/PKDD Workshop on Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery. Warsaw 2007.
7. Svátek V., Rauch J., Ralbovský M.: Ontology-Enhanced Association Mining. In: Ackermann, Berendt (eds.). Semantics, Web and Mining, Springer-Verlag, 2006.
8. Zeman, M., Ralbovský M., Svátek V., Rauch J.: Ontology-Driven Data Preparation for Association Mining. Online `http://keg.vse.cz/onto-kdd-draft.pdf`.

[7] This was influenced by the concept-centric nature of UMLS; for a more DB-schema-like ontology, `owl:dataProperty` or its new subclass would fit well, in turn.