Debugging classical ontologies using defeasible reasoning tools (extended abstract)

Simone Coetzer and Katarina Britz

CAIR, Stellenbosch Univ., South Africa
scoetzer50@gmail.com; abritz@sun.ac.za

Ontologies provide knowledge engineers with the ability to represent and encode knowledge in a formal language so that it can be processed or 'reasoned over' by a computer [7]. Notable benefits include the ability to source new knowledge by making statements that are implicitly deduced explicitly available to the end-user, to classify individuals or instances and to check the addition of new knowledge for logical consistency [1].

Given the nature and goal of ontologies, a successful application of ontologies relies on (1) representing as much relevant domain knowledge accurately (2) while maintaining logical consistency. As the successful implementation of a real-world ontology is likely to contain many concepts and intricate relationships between the concepts, it is necessary to follow a methodology for debugging the ontology [5]. A myriad of ontology debugging approaches (some of them instantiated in tools) have been developed to help the knowledge engineer pinpoint the cause of logical inconsistencies and rectify them in a strategic way [6, 3, 4]. Usually, a Model-Based Diagnosis approach is followed to debug the ontology: this involves finding the *diagnosis* which contains all minimal conflict sets; the diagnosis is then presented to the knowledge engineer who will need to amend the axioms by modifying or deleting certain axioms; once the ontology has been amended, it is again checked whether an inconsistency is present - if it is, the process is repeated [5].

Although most ontology debugging approaches localise the faulty axioms, they do not (to date) provide recommendations on how logical inconsistencies can be resolved by weakening (instead of deleting) faulty axioms. We propose a theoretical methodology for weakening faulty axioms in a strategic way using defeasible reasoning tools. Our methodology draws from Rodler's [5] interactive ontology debugging approach which not only localises faulty axioms but provides the knowledge engineer with a strategic way of resolving them by presenting the root cause inconsistencies first. We are extending this approach by suggesting that through the use of defeasible reasoning techniques, a methodology can be created to systematically find conflict resolution recommendations.

Importantly, our goal is not to convert a classical ontology to a defeasible ontology - therefore we do not use defeasible reasoning support through, for example, the computation of rational closure. Rather, we use the definition of exceptionality of a concept, which is central to the semantics of defeasible DLs, and the associated algorithm (as can be found in [2]) to determine the extent of a concept's exceptionality (their ranking); then, starting with the statements containing the most general concepts (the least exceptional concepts) weakened versions of the original statements are constructed; this is done until all inconsistencies have been resolved. Consider how this methodology can be applied to the following set of statements in an example ontology:

$User \sqsubseteq \neg\exists AccessTo.ConfidentialInfo$
$Staff \sqsubseteq User$
$Staff \sqsubseteq \exists AccessTo.ConfidentialInfo$
$BlackListedStaff \sqsubseteq Staff$
$BlackListedStaff \sqsubseteq \neg\exists AccessTo.ConfidentialInfo$

When running the ranking algorithm, the concept 'User' is the most general exceptional concept; the concept 'Staff' is more exceptional than the concept of 'User'. Starting with the least exceptional concept, 'User', we then weaken statements containing this concept on the left hand side of the axiom by using a conjunction between the current concept under investigation and the negation of the concept associated with it on the next level of exceptionality (Staff) - in this phase, the ontology is transformed as follows:

$User \sqcap \neg Staff \sqsubseteq \neg\exists AccessTo.ConfidentialInfo$
$Staff \sqsubseteq User$
$Staff \sqsubseteq \exists AccessTo.ConfidentialInfo$
$BlackListedStaff \sqsubseteq Staff$
$BlackListedStaff \sqsubseteq \neg\exists AccessTo.ConfidentialInfo$

The ranking of all concepts is again calculated and the first exceptional concept, 'User' has now been resolved. Then, we move onto the next concept, Staff, which is now the most general exceptional concept. The same kind of transformation is performed on axioms containing 'Staff' on the left hand side (with the exclusion of axioms containing one of the previous exceptional statements on the right hand side):

$User \sqcap \neg Staff \sqsubseteq \neg\exists AccessTo.ConfidentialInfo$
$Staff \sqcap \sqsubseteq User$
$Staff \sqcap \neg BlackListedStaff \sqsubseteq \exists AccessTo.ConfidentialInfo$
$BlackListedStaff \sqsubseteq Staff$
$BlackListedStaff \sqsubseteq \neg\exists AccessTo.ConfidentialInfo$

The forward-looking goal of this research is to provide a methodological foundation which could, in future, lead to the development and implementation of an inconsistency resolution recommender tool that is fully integrated with the ontology development environment.

References

1. Baader, F., Horrocks, I. and Sattler, U., Description Logics, *in* Staab, S. and Studer, R., eds, Handbook On Ontologies, Springer-Verlag, Berlin, Chapter 1, pp. 3-28, 2004.
2. Britz, K., Casini, G., Meyer, T. and Varzinczak, I., A KLM perspective on defeasible reasoning for description logics, *in* Lutz, C. and Sattler, U. and Tinelli, C. and Turhan, A.-Y. and Wolter, F., eds, Description Logic, Theory Combination, and All That. Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday, pp. 147-173, 2019.
3. Friedrich, G. and Schekotykhin, K., A General Diagnosis Method for Ontologies, Proceedings of the 4th International Semantic Web Conference (ISWC 2005), 2005.
4. Kalyanpur, A., Parsia, B., Horridge, M. and Sirin, E., Finding all Justifications of OWL DL Entailments, The Semantic Web, 6th International Semantic Web Conference, 2007.
5. Rodler, P., Interactive Debugging of Knowledge Bases, PhD thesis: Alpen-Adria University Klagenfurt, 2015.
6. Schlobach, S., Huang, Z., Cornet, R. and Harmelen, F., Debugging Incoherent Terminologies, Journal of Automated Reasoning, Volume 39, pp. 317-349, 2007.
7. Studer, R., Benjamins, V.R. and Fensel, D., Knowledge Engineering: Principles and Methods, Data and Knowledge Engineering, Volume 25, pp. 161-197, 1998.