2009

RaDON - Repair and Diagnosis in Ontology Networks

Qiu Ji

Peter Haase

Guilin Qi

Pascal Hitzler
pascal.hitzler@wright.edu

Steffen Stadtmuller

RaDON — Repair and Diagnosis in Ontology Networks*

Qiu Ji, Peter Haase, Guilin Qi, Pascal Hitzler, Steffen Stadtmüller

Institute AIFB
Universität Karlsruhe (TH), Germany
qiji,pha,gqi,phi@aifb.uni-karlsruhe.de,
steffen.stadtmueller@stud.kit.edu

Abstract. One of the major challenges in managing networked and dynamic ontologies is to handle inconsistencies in single ontologies, and inconsistencies introduced by integrating multiple distributed ontologies. Our RaDON system provides functionalities to repair and diagnose ontology networks by extending the capabilities of existing reasoners. The system integrates several new debugging and repairing algorithms, such as a relevance-directed algorithm to meet the various needs of the users.

1 Introduction

Next generation semantic applications are characterized by a large number of ontologies, some of them constantly evolving. As the complexity of semantic applications increases, more and more knowledge is embedded in applications, typically drawn from a wide variety of sources. This new generation of applications thus likely relies on ontologies embedded in a network of other ontologies. Ontologies and metadata have to be kept up to date when application environments and users' needs change. One of the major challenges in managing these networked and dynamic ontologies is to handle potential inconsistencies in single ontologies, and inconsistencies introduced by integrating multiple distributed ontologies.

For inconsistency handling in single, centralized ontologies, several approaches are known, see the survey in [4, 3]. There are mainly two ways to deal with inconsistent ontologies [5]. One way is to simply live with the inconsistency and to apply a non-standard reasoning method to obtain meaningful answers. The second way to deal with logical contradictions is to resolve logical modeling errors whenever a logical problem is encountered [16, 15, 12].

There is relatively little work done on handling inconsistency in networked ontologies. For example, in [1], the authors deal with the problem of inconsistency in DDL by removing some bridge rules which are responsible for the inconsistency. However, there is no tool available to diagnose and repair inconsistencies in networked ontologies.

To meet the above mentioned needs, we develop the RaDON system to repair and diagnose not only single ontologies but networked ones, where several new debugging algorithms, such as a relevance-directed algorithm and a paraconsistency-based algorithm, have been integrated into RaDON to meet the various needs of the users.

* This work is partially supported by the EU in the IST project NeOn (IST-2006-027595).

2 The RaDON System

RaDON[1] has been developed to deal with inconsistency[2] and incoherence[3] for ontology networks. It supports OWL-DL and is implemented in Java as a plug-in for the NeOn toolkit[4], which is an extensible ontology engineering environment to handle multiple networked ontologies. The RaDON plugin has already been applied in the FAO[5] case study in the context of diagnosing and repairing automatically learned ontologies. Results of these applications have been reported in NeOn Deliverable D1.2.2 [13].

In this section, we first introduce the functionalities of the RaDON system and then we present the process to debug and repair ontology networks. Afterwards, we describe the two new algorithms respectively: relevance-directed algorithm and paraconsistency-based algorithm for debugging ontologies.

2.1 Functionalities of RaDON

RaDON provides a set of techniques for dealing with inconsistency and incoherence in ontologies. In particular, RaDON supports novel strategies and consistency models for distributed and networked environments.

RaDON extends the capabilities of existing reasoners with the functionalities to deal with inconsistency and incoherence. Specifically, the functionalities provided by RaDON include: (1) debugging an incoherent or inconsistent ontology to explain why a concept is unsatisfiable or why the ontology is inconsistent, (2) repairing an ontology automatically by computing all possible explanations w.r.t. all unsatisfiable concepts if the ontology is incoherence, or w.r.t. the inconsistent ontology if it is inconsistent, (3) repairing an ontology manually based on the debugging results. For the manual repair, the user can choose the axioms to be removed for restoring the coherence or consistency. (4) coping with inconsistency based on a paraconsistency-based algorithm.

2.2 Process to Debug and Repair Ontology Networks

The plug-in can be used to diagnose and repair not only a single ontology, but also multiple ontologies that are networked. In particular, we consider ontologies that are networked via *mappings*. Mappings essentially are correspondences between the elements of two different ontologies, in the most simple case in the form of subclassOf or equivalentClasses axioms (cf. [2] for a definition of the networked ontology model). The mapping assertions may additionally be annotated with confidence values.

RaDON takes as input an ontology network consisting of ontologies and mappings between the ontologies. For our definition of inconsistency in the ontology network, we follow a global model semantics: An ontology network is inconsistent (resp. incoherent)

[1] http://radon.ontoware.org/demo.htm

[2] An ontology is inconsistent iff it has no model.

[3] An ontology is incoherent iff it contains at least one unsatisfiable concept in its terminology. A concept is unsatisfiable if it is mapped to the empty set in all models of the ontology.

[4] http://www.neon-toolkit.org/

[5] http://www.fao.org/

if the ontology obtained by merging the ontologies and their mappings is inconsistent (resp. incoherent). For the repair process we focus on repairing the mappings, assuming the ontologies are already individually coherent and consistent. Consequently only axioms in the mappings could be removed to resolve incoherence or inconsistency.

If the merged ontology is incoherent, the system computes all the unsatisfiable concepts. When an unsatisfiable concept is selected, all the minimal unsatisfiability-preserving subsets (MUPS)[6] w.r.t. this concept are computed. Similarly, inconsistencies in the ontology network are debugged: For inconsistencies, all the minimal inconsistent subsets (MIS)[7] w.r.t. the ontology are computed.

After obtaining the debugging results, we can repair the ontology automatically according to the proposed solution by our system. Also, the user can repair the ontology by manually choosing the axioms to be removed. To help the user to make a decision, our system provides the confidence values or scores for each axiom which could be removed to resolve incoherence or inconsistency. See [13] for more details.

2.3 Relevance-directed Algorithm

A key problem of diagnosing and repairing ontology networks is to compute all or some MUPS for an unsatisfiable concept efficiently. The RaDON system provides various strategies to compute some or all MUPS based on the relevance-directed algorithm [7]. This algorithm incrementally selects sub-ontologies using a *selection function* and finds a set of MUPS from these sub-ontologies for an unsatisfiable concept. Our algorithm is adapted from the algorithm given in [8] which is based on Reiter's Hitting Set Tree (HST) algorithm [14].

Specifically, the relevance-directed algorithm provides the following strategies when computing MUPS for an unsatisfiable concept:

- Compute one MUPS;
- Compute all MUPS and all hitting sets[8] for the set of all MUPS;
- Compute some (not all) MUPS and some hitting sets for the set of all MUPS.

Therefore, the users could choose different strategies according to the ontology and their purpose. For example, if the testing ontology is relatively small, they could try to compute all MUPS. Also, if the users only intend to resolve the incoherence with some but not all solutions, we can choose the third strategy.

2.4 Paraconsistency-based Algorithm

In RaDON we also provide the functionality to provide inconsistency-tolerant reasoning. Specifically, an algorithm for paraconsistent reasoning given in [10] has been integrated into RaDON which helps the user to cope with the inconsistency. This algorithm

[6] Let C be a named concept which is unsatisfiable in an ontology $\mathcal{O}$. A set $\mathcal{M}\subseteq\mathcal{O}$ is a *minimal unsatisfiability-preserving subset (MUPS)* [16] of $\mathcal{O}$ if C is unsatisfiable in $\mathcal{M}$, and C is satisfiable in every subset $\mathcal{M}' \subset \mathcal{M}$.

[7] Given an inconsistent ontology $\mathcal{O}$, a set $\mathcal{M}\subseteq\mathcal{O}$ is a *minimal inconsistent subset (MIS)* w.r.t. $\mathcal{O}$ if $\mathcal{M}$ is inconsistent and every subset $\mathcal{M}' \subset \mathcal{M}$ is consistent.

[8] Given a set $S = \{\mathcal{M}_1, ..., \mathcal{M}_n\}$ of MUPS of an ontology $\mathcal{O}$ for an unsatisfiable concept, a *hitting set* T for S is a subset of $\mathcal{O}$ such that $\mathcal{M}_i \cap T \neq \emptyset$ for all $1 \leq i \leq n$.

considers inconsistent ontologies to be based on a four-valued semantics. Such ontologies are translated to new consistent ontologies, in a way such that classical two-valued reasoning tasks with the translated ontologies result in the same conclusions as paraconsistent four-valued reasoning. The user can decide whether he wants to translate all inconsistent ontologies automatically or select the ontologies, that should be translated. Further information on the theory can be found in [10].

3 Related Work

To our best of knowledge, the most closely related tools are Swoop, Protégé, PION and DION.

Swoop [9] provides a user interface for computing explanations of an unsatisfiable concept and for repairing an ontology using several kinds of ranking methods. Our plug-in differs from theirs in the following main aspects. First of all, Swoop only considers a single ontology while RaDON can deal with not only single ontologies but ontology networks. Secondly, RaDON can repair both incoherence and inconsistency. Thirdly, the internal repair algorithm by using confidence values is different (more details can be found in [13]). Finally, we integrated our plug-in into the NeOn toolkit which provides a flexible framework to deal with ontology networks.

As for Protégé [11] (here, we use Protégé 4.0 alpha), it is able to compute explanations and provide a manual way to repair an ontology. The differences mentioned above for Swoop can apply to Protégé as well. Furthermore, no functionality to repair an ontology automatically is implemented in Protégé.

PION [6] and DION [17] have been developed in the SEKT project[9] to deal with inconsistency. PION is an inconsistency-tolerant reasoner which can return meaningful answers when querying an inconsistent ontology. It tries to answer a query by selecting a consistent sub-ontology from the inconsistent ontology. DION provides debugging support to compute MUPS or MIPS. But DION does not consider repairing inconsistent or incoherent ontologies. Furthermore, DION it cannot be applied to deal with very expressive ontologies, such as OWL DL ontologies.

4 Conclusion and Future Work

In our demonstration, we present the RaDON system, which provides support to diagnose and repair ontologies automatically or manually. In particular, RaDON supports novel strategies to handle incoherence and inconsistency for ontology networks. To repair an ontology network in which the ontologies are related via mappings, we focus on the repair of mappings between ontologies, assuming the individual ontologies are locally coherent and consistent and are more important and reliable than the mapping. In the demonstration, the process to debug and repair ontology networks will be shown with practical examples. Besides, for various needs of the users different algorithms to deal with incoherence and inconsistency will be demonstrated.

[9] http://www.sekt-project.com/project

As future work, we intend to support ontology networks in which ontologies are networked in different way, e.g. via dependency and extension relationships in modular ontologies.

References

1. A. T. Christian Meilicke, Heiner Stuckenschmidt. Repairing ontology mappings. In *Proceeding of 22th National Conference on Artificial Intelligence (AAAI)*, pages 1408–1413. 2007.
2. P. Haase, S. Brockmans, R. Palma, J. Euzenat, and M. d'Aquin. D1.1.2 updated version of the networked ontology model. NeOn Project Deliverable D1.1.2, Universität Karlsruhe (TH), 2007.
3. P. Haase and G. Qi. An analysis of approaches to resolving inconsistencies in DL-based ontologies. In *Proceeding of International Workshop on Ontology Dynamics(IWOD)*, 2007.
4. P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *Proceeding of 4th International Semantic Web Conference (ISWC)*, pages 353–367, 2005.
5. Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *Proceeding of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 254–259. 2005.
6. Z. Huang, F. van Harmelen, A. ten Teije, P. Groot, and C. Visser. D3.4.1 reasoning with inconsistent ontologies: a general framework. SEKT Project Deliverable D3.4.1, Vrije Universiteit Amsterdam, 2004.
7. Q. Ji, G. Qi, and P. Haase. A relevance-based algorithm for finding justifications of DL entailments. In *Technical report, University of Karlsruhe, http://www.aifb.uni-karlsruhe.de/WBS/gqi/papers/RelAlg.pdf*, 2008.
8. A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin. Finding all justifications of OWL DL entailments. In *Proceeding of 6th International and 2nd Asian Semantic Web Conference (ISWC/ASWC)*, pages 267–280, 2007.
9. A. Kalyanpur, B. Parsia, E. Sirin, B. C. Grau, and J. A. Hendler. Swoop: A web ontology editing browser. *Journal of Web Semantics*, 4(2):144–153, 2006.
10. Y. Ma, P. Hitzler, and Z. Lin. Algorithms for paraconsistent reasoning with OWL. In *Proceeding of 4th European Semantic Web Conference (ESWC)*, pages 399–413, 2007.
11. N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.
12. B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proceeding of 14th International World Wide Web Conference (WWW)*, pages 633–640, 2005.
13. G. Qi, P. Haase, Q. Ji, and J. Völker. D1.2.2 consistency models for networked ontologies evaluation. NeOn Project Deliverable D1.2.2, Universität Karlsruhe, 2007.
14. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
15. S. Schlobach. Diagnosing terminologies. In *Proceeding of 20th National Conference on Artificial Intelligence (AAAI)*, pages 670–675, 2005.
16. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proceeding of 18th International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 355–362, 2003.
17. S. Schlobach and Z. Huang. D3.6.1 inconsistent ontology diagnosis: Framework and prototype. SEKT Project Deliverable D3.6.1, Vrije Universiteit Amsterdam, 2005.