On the Feasibility of Using OWL 2 Reasoners in Ontology Alignment Repair Problems

Alessandro Solimando[1], Ernesto Jiménez-Ruiz[2], and Giovanna Guerrini[1]

[1] DIBRIS, Università di Genova, Italy
[2] Department of Computer Science, University of Oxford, UK

1 Introduction

The problem of (semi-)automatically computing mappings between independently developed ontologies is usually referred to as the ontology matching problem. A number of sophisticated ontology matching systems have been developed in the last years [7, 26]. Ontology matching systems, however, rely on lexical and structural heuristics and the integration of the input ontologies and the mappings may lead to many undesired logical consequences (*e.g.*, unsatisfiable classes).

The fix of unsatisfiable classes caused by ontology mappings is known as the mapping repair problem [13]. Mapping repair can be addressed using state-of-the-art approaches for debugging inconsistencies in OWL 2 ontologies, which rely on the extraction of justifications for the unsatisfiable classes (*e.g.*, [24, 14, 29, 12]). However, in [10] it was pointed out that justification-based technologies do not scale when the number of unsatisfiabilities is large (a typical scenario in mapping repair problems).

In this paper we provide an update on the results presented in [10] by evaluating the feasibility of using up-to-date OWL 2 reasoners in mapping repair problems. We have conducted an extensive evaluation using the datasets and ontology matching systems from the Ontology Alignment Evaluation Initiative (OAEI) [7]. Our results suggest that the classification of the integration of large ontologies via mappings still poses a challenge to OWL 2 reasoners. Furthermore, the repair of unintended entailments (*e.g.*, unsatisfiable concepts) using OWL 2 reasoners critically compromises the performance of mapping repair systems.

2 Preliminaries

In this section, we present the formal representation of ontology mappings and the notions of semantic difference and mapping coherence.

Representation of Ontology Mappings. Mappings are conceptualised as 4-tuples of the form $\langle e_1, e_2, n, \rho \rangle$, with e_1, e_2 entities in the vocabulary or signature of the relevant input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ (*i.e.*, $e_1 \in \mathsf{Sig}(\mathcal{O}_1)$ and $e_2 \in \mathsf{Sig}(\mathcal{O}_2)$), n a confidence measure between 0 and 1, and ρ a relation between e_1 and e_2, typically subsumption, equivalence or disjointness [6].

RDF Alignment [4] is the main format used in the Ontology Alignment Evaluation Initiative (OAEI) to represent mappings containing the aforementioned elements. Additionally, mappings are also represented as OWL 2 subclass, equivalence, and disjointness axioms [2]; mapping confidence values (n) are then represented as axiom annotations. Such a representation enables the reuse of the extensive range of OWL 2 reasoning infrastructure that is currently available. Note that alternative formal semantics for ontology mappings have been proposed in the literature (*e.g.*, [1]).

Mapping Coherence and Mapping Repair. The ontology resulting from the integration of $\mathcal{O}_1$ and $\mathcal{O}_2$ via a set of mappings $\mathcal{M}$ typically entails axioms that do not follow from $\mathcal{O}_1$, $\mathcal{O}_2$, or $\mathcal{M}$ alone. Some of these axioms may represent undesired entailments, due to erroneous mappings in $\mathcal{M}$ or to inherent incompabilities between the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and may lead to unsatisfiable classes.

A set of mappings that leads to unsatisfiable classes in $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M}$ is referred to as *incoherent* w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$ [18].

Definition 1 (Mapping Incoherence). *A set of mappings $\mathcal{M}$ is incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$, if there exists a class A in the signature of $\mathcal{O}_1 \cup \mathcal{O}_2$ such that $\mathcal{O}_1 \cup \mathcal{O}_2 \not\models A \sqsubseteq \bot$ and $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{M} \models A \sqsubseteq \bot$.*

An incoherent set of mappings $\mathcal{M}$ can be fixed by removing mappings from $\mathcal{M}$. This process is referred to as *mapping repair* (or repair for short).

Definition 2 (Mapping Repair). *Let $\mathcal{M}$ be an incoherent set of mappings w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$. A set of mappings $\mathcal{R} \subseteq \mathcal{M}$ is a mapping repair for $\mathcal{M}$ w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$ if $\mathcal{M} \setminus \mathcal{R}$ is coherent w.r.t. $\mathcal{O}_1$ and $\mathcal{O}_2$.*

A trivial repair is $\mathcal{R} = \mathcal{M}$, since an empty set of mappings is obviously coherent. Nevertheless, the objective is to remove as few mappings as possible. Minimal (mapping) repairs are typically referred to in the literature as *mapping diagnosis* [17] — a term coined by Reiter [22] and introduced to the field of ontology debugging in [25].

Definition 3 (Mapping diagnosis). *Let $\mathcal{R}$ be a repair for $\mathcal{M}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. $\mathcal{R}$ is a diagnosis if each $\mathcal{R}' \subset \mathcal{R}$ is not a repair for $\mathcal{M}$ with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$.*

In the literature there are different approaches to compute a repair or diagnosis for an incoherent set of mappings. Early approaches were based on Distributed Description Logics (DDL) (*e.g.*, [19, 20, 21]). Alternatively, if mappings are represented as OWL 2 axioms, a repair or diagnosis can also be computed using the state-of-the-art approaches for debugging and repairing inconsistencies in OWL 2 ontologies, which rely on the extraction of justifications for the unsatisfiable classes (*e.g.*, [24, 14, 29, 12]). In ontology matching scenarios is very frequent the use of incomplete reasoning techniques to enhance scalability (*e.g.*, [11, 17, 23]). Incomplete reasoning leads to an *approximate repair* $\mathcal{R}^{\approx}$, *i.e.*, there is no guaranteee that $\mathcal{M} \setminus \mathcal{R}^{\approx}$ is coherent.

3 Evaluation

This section describes the conducted experimental evaluation. In Section 3.1 we present the datasets and used mapping sets. Section 3.2 introduces the evaluation setting. The obtained results are discussed in Section 3.3.

Table 1: Metrics about the ontologies employed in the evaluation.

Ontology	**Track**	**#Concepts**	**#DatatypeP.**	**#ObjectP.**	**DL**
CMT	Conference	36	10	49	$\mathcal{ALCIN}(\boldsymbol{D})$
CONFERENCE	Conference	60	18	46	$\mathcal{ALCHIF}(\boldsymbol{D})$
CONFOF	Conference	38	23	13	$\mathcal{SIN}(\boldsymbol{D})$
EKAW	Conference	74	0	33	$\mathcal{SHIN}$
IASTED	Conference	140	3	38	$\mathcal{ALCIN}(\boldsymbol{D})$
SIGKDD	Conference	49	11	17	$\mathcal{ALEI}(\boldsymbol{D})$
FMA (NCI)	Largebio	3696	24	0	$\mathcal{ALCN}(\boldsymbol{D})$
FMA (SNOMED)	Largebio	10157	24	0	$\mathcal{ALCN}(\boldsymbol{D})$
NCI (FMA)	Largebio	6488	0	63	$\mathcal{ALC}$
NCI (SNOMED)	Largebio	23958	0	82	$\mathcal{ALCH}$
SNOMED (FMA)	Largebio	13412	0	18	$\mathcal{ALER}$
SNOMED (NCI)	Largebio	51128	0	51	$\mathcal{ALER}$
STW	Library	6575	0	0	$\mathcal{AL}$
TheSoz	Library	8376	0	0	$\mathcal{AL}$

3.1 Datasets

The datasets are based on the OAEI, an international campaign for the systematic evaluation of ontology matching systems. The matching problems in the OAEI are organised in several tracks, with each track involving different kinds of test ontologies[7, 3, 5]. In this paper we have focused on the *largebio*, *library* and *conference* tracks. For largebio we used fragments of FMA, NCI and SNOMED CT, because they already posed challenges to the reasoners. Note that the used fragments represent relevant portions of one of the ontologies with respect to the other two. For example, the fragment of FMA relevant to NCI contains 3,696 concepts (see Table 1). Library is composed by not very expressive medium-sized ontologies, while conference ontologies are usually very expressive but of limited size. Table 1 summarizes the metrics of the selected ontology pairs for the evaluation, while Table 2 provides the details about the selected subset of mapping sets computed by ontology matching systems participating in the OAEI 2013 and 2014 campaigns.[3] Please refer to [3, 5] for more information about the datasets and ontology matching systems.

3.2 Evaluation Settings

System Details. The test environment consists of a desktop computer equipped with 32*GB DDR*3 *RAM* at 1333*MHz*, and an *AMD Fusion FX 4350* (quad-core, each running at 4.2*GHz*) as *CPU*. The dataset is stored on a 128*GB SSD*, where the operating system (*Ubuntu* 12.04, 64-bit version) is installed. The employed build of Java Runtime Environment (JRE) is 1.8.0_45-b14, while the one for the Oracle 64-Bit Java Virtual Machine (JVM) is the 25.45-b02 (mixed mode). The amount of memory allocated for the heap of the JVM is 12GB, the processes not involved in the evaluation require approximately 3GB of space, thus leaving 17GB of free RAM (plus 1.8GB of swap memory, that is not used unless totally necessary[4]).

---

[3] Due to space and time reasons we selected only a subset of the computed mappings sets we considered representative.

[4] This behaviour is enforced by means of the swappiness Linux kernel parameter set to 0, see `http://en.wikipedia.org/wiki/Swappiness` for more information.

Table 2: Metrics about the mapping sets employed in the evaluation.

Ontology 1	Ontology 2	# Mappings	Matching System
FMA	NCI	5960	MaasMatch$_{14}$
FMA	NCI	5781	LogMapBio$_{14}$
SNOMED	NCI	2500	IAMA$_{13}$
SNOMED	NCI	3040	OMReasoner$_{14}$
SNOMED	NCI	13270	YAM++$_{13}$
SNOMED	NCI	13582	AML$_{14}$
FMA	SNOMED	21110	GOMMA$_{13}$
FMA	SNOMED	16812	IAMA$_{13}$
FMA	SNOMED	28262	AML$_{14}$
FMA	SNOMED	28711	LogMapBio$_{14}$
FMA	SNOMED	23344	YAM++$_{13}$
IASTED	SIGKDD	70	AOTL$_{14}$
CONFOF	IASTED	10	AML$_{14}$
CONFERENCE	EKAW	164	MaasMatch$_{14}$
CMT	IASTED	32	MaasMatch$_{14}$
CONFERENCE	IASTED	68	MaasMatch$_{14}$
STW	TheSoz	7254	AML$_{14}$
STW	TheSoz	12032	Hertuda$_{13}$
STW	TheSoz	378	IAMA$_{13}$
STW	TheSoz	5684	LogMap$_{13}$
STW	TheSoz	342	RSDLWB$_{14}$
STW	TheSoz	80686	XMapGen$_{13}$
STW	TheSoz	2870	XMapSig$_{13}$

Tested Reasoners. The versions of the employed reasoner are: *(i) Konclude* 0.6.0-408 64-bit [28] *(ii) ELK* 0.4.1 [16] *(iii) Pellet* 2.3.1 [27] *(iv) HermiT* 1.3.8 [8].

ELK, *Pellet* and *HermiT* implement the *OWLReasoner* interface of the *OWL-API* and they all are called on a fresh thread. A timeout on the classification task is enforced by killing the thread after reaching the timeout value, times are measured using the *getNanoSec* function, because it measures the elapsed time without skew corrections.[5]

ELK is a (very fast) reasoner for the OWL 2 EL profile, thus it cannot guarantee complete results for ontologies outside this profile.

Konclude does not implement the *OWL-API*'s *OWLReasoner* interface and its invocation through *OWLlink* 1.2.1 is raising an *OWLlinkReasonerRuntimeException* exception caused by an *IndexOutOfBoundsException* exception during the parsing of most of the ontologies in our dataset. Thus, *Konclude* is instead called using an external process,[6] using the *ProcessBuilder* class,[7] and it is allowed to use all the available cores. For *Konclude*, timeout on classification is enforced using *timeout* program for Linux, and wall-clock time is measured using the *time* program.[8]

It was not possible to extend our analysis to *FaCT++* 4.3 because its invocation using *JNI* is permanently failing with a *StackOverflowError*.

Justification Extractor. In this paper we have used the *black-box* justification extractor described in [9].[9]. Black box extractors typically allow to use any reasoner implement-

---

[5] `https://docs.oracle.com/javase/8/docs/api/java/lang/System.html#nanoTime--`

[6] *Konclude* is runned with "Konclude classification -w AUTO -i aligneOntology.owl"

[7] `https://docs.oracle.com/javase/8/docs/api/java/lang/ProcessBuilder.html`

[8] Using "/usr/bin/time -f %E cmd" command.

[9] Current version available at `https://github.com/matthewhorridge/owlexplanation`. For the experiments we used the version available here:

Table 3: Classification times (s) in largebio dataset with selected mapping sets.

Dataset / Reasoner	FMA-NCI		FMA-SNOMED		SNOMED-NCI		
	MaasMatch_{14}	LogMapBio_{14}	YAM++_{13}	AML_{14}	AML_{14}	GOMMA_{13}	YAM++_{13}
ELK	0.21	0.08	0.6	0.3	3.1	2.91	3.44
HERMIT	3.32	20.19	5.08	10.49	T/OUT	49	T/OUT
KONCLUDE	1.3	8.25	3.83	4.82	OOM	OOM	OOM
PELLET	T/OUT	30.46	T/OUT	2198.82	T/OUT	T/OUT	T/OUT

ing the axiom pinpointing service. In the future we also plan to evaluate *glass-box* justification techniques as the implemented in Pellet or ELK [15].

Note that *Konclude*, since it was invoked from the command line, could not be evaluated on the justification extraction tasks.

3.3 Experimental Evaluation

We have conducted the following evaluation. We take as input a pair of input ontologies ($\mathcal{O}_1$ and $\mathcal{O}_2$) and an alignment $\mathcal{M}$ between them from the datasets described in Section 3.1. For each of the available reasoners we compute the classification[10] and record the classification times in seconds (see Tables 3-5 and *Class.(s)* in Tables 6-9). Then, if the classification succeeds, we record the number of unsatisfiable concepts (*#Unsat* in Tables 6-9) and, for at most 50 of them, we compute justifications[11] (a single one and up to a maximum of 10 justifications[12]), recording the total time in seconds required for completing the respective operations (*1Just.(s)* and *10Just.(s)* in Tables 6-9, respectively).

Classification. In Tables 3-5 the classification time for a selection of the testcases is shown. *Pellet* failed to classify, due to timeouts (T/OUT), most of the largebio aligned ontologies, as shown in Table 3. Only *ELK* could classify the integration of SNOMED and NCI in most of the cases.[13] *Konclude*, for instance, failed with an out of memory error (OOM). For library, instead, the reasoners succeeded in most of the cases, but only *Konclude* managed to classify, within the timeout, the integrated ontology via the mappings computed by XMapGen. These mappings include an extraordinary number of many to many correspondences, that caused problems to all the reasoners but *Konclude*. Concerning conference (Table 5), the classification could be performed in the vast majority of the cases, with only a single failure for both *HermiT* and *Pellet*.

Computation of Justifications. Tables 6-9, instead, show the details for justification computation for relevant cases. Library results are omitted due to the lack of unsatisfiable classes in the aligned ontologies (the input ontologies are simple and they do not contain disjointness axioms).

`https://github.com/protegeproject/mvn-repo/tree/master/releases/org/semanticweb/owl/explanation/3.3.0`

[10] With a timeout of 60, 20 and 10 minutes for largebio, library and conference, respectively.

[11] With a timeout of 60 seconds to find each new justification.

[12] Extracting 10 justification is already rather time consuming; nevertheless, in future evaluations, we plan to extend the limit up to 50 justifications.

[13] Note that ELK is an OWL 2 EL reasoner and since NCI falls outside the OWL 2 EL profile, the classification computed by ELK for the integration of SNOMED and NCI is incomplete.

Table 4: Classification times (s) in library dataset with selected mapping sets.

Dataset / Reasoner	STW-TheSoz						
	AML$_{14}$	Hertuda$_{13}$	IAMA$_{13}$	LogMap$_{13}$	RSDLWB$_{14}$	XMapGen$_{13}$	XMapSig$_{13}$
ELK	0.73	45	0.24	0.25	0.13	T/OUT	0.25
HERMIT	4.82	842	1.08	2.23	1.14	T/OUT	1.7
KONCLUDE	2.28	17	1.13	1.72	1.2	59	1.77
PELLET	8.7	T/OUT	0.21	1.42	0.45	T/OUT	0.92

Table 5: Classification times (s) in conference dataset with selected mapping sets.

Dataset / Reasoner	**CMT-IASTED**	**CONFERENCE-IASTED**	**CONFOF-IASTED**	**IASTED-SIGKDD**
	MaasMatch$_{14}$	MaasMatch$_{14}$	AML$_{14}$	AOTL$_{14}$
ELK	0.01	0.01	0	0.01
HERMIT	0.22	T/OUT	0.28	24
KONCLUDE	0.09	0.36	0.12	0.25
PELLET	T/OUT	10	4.76	23

Table 6: Justification extraction in the FMA-NCI largebio dataset

(a) With MaasMatch$_{14}$

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	0.21	7,377	15	162
HERMIT	3.32	8,767	43	1,206
PELLET	T/OUT	-	-	-

(b) With LogMapBio$_{14}$

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	0.08	0	0	0
HERMIT	20	467	15	863
PELLET	30	467	11	493

Table 7: Justification extraction in the FMA-SNOMED largebio dataset

(a) With IAMA$_{13}$

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	0.41	22,925	9.74	55
HERMIT	0.58	22,925	5.11	30
PELLET	1.78	22,925	4.84	14

(b) With OMReasoner$_{14}$

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	0.44	478	12	85
HERMIT	10	478	6.73	43
PELLET	195	478	5.41	23

Note that, the computed times in the Tables 6-9 are only for 50 unsatisfiable classes. Thus, the total times given below for all unsatisfiable classes have been extrapolated from these results.

Consider Table 6a which presents the justification extraction results for the integration of FMA and NCI via the mappings computed by MaasMatch. Computing a single justification for each unsatisfiable concept (7,377) would require for *ELK* $>$36m (15s for 50 unsatisfiable classes), while $>$6h for computing ten of them (162s for 50 unsat classes). When *HermiT* is used, $>$2h and $>$58h would be required, respectively.

In Tables 8a-8b, the values are definitely higher. Computing a single justification for each unsatisfiable concept in the testcase of Table 8a would require, for *ELK* (resp. *HermiT*), $>$12h (resp. $>$11h), while $>$72h (resp. $>$16 days) for computing ten of them.

Considering small sized ontologies, but with high expressivity, we also find cases that could not be compatible with an "online" mapping repair (*e.g.*, $>$30m for *HermiT* in Table 9a, and $>$28m for *Pellet* in Table 9b).

Table 8: Justification extraction in the SNOMED-NCI largebio dataset

(a) With GOMMA_{13}

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	2.91	50,189	45	259
HERMIT	49	53,448	39	1,350
PELLET	T/OUT	-	-	-

(b) With IAMA_{13}

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	2.37	40,002	35	119
HERMIT	56	44,017	38	584
PELLET	T/OUT	-	-	-

Table 9: Justification extraction in the conference dataset

(a) IASTED-SIGKDD with AOTL_{14}

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	0.01	4	0.58	13
HERMIT	24	5	46	1,853
PELLET	23	5	11	274

(b) Conference-EKAW with MaasMtch_{14}

Reasoner	**Class.(s)**	**#Unsat**	**1Just.(s)**	**10Just.(s)**
ELK	0.03	54	7.9	51
HERMIT	0.03	63	5.31	86
PELLET	0.02	63	2.37	1,354

4 Conclusions

In this paper we have evaluated the feasibility of using OWL 2 reasoning capabilities in mapping repair related tasks. For this purpose, we have evaluated the performances of several top-level reasoners on classification and justifications computation. Our empirical results suggest that the classification of the integration of medium/large size ontologies via mappings, although feasible, still poses serious problems to current OWL 2 reasoners. Furthermore, when OWL 2 reasoners are to be used in mapping repair tasks, the computation time increases considerably, and in most cases it is simply impractical, even when using (scalable but incomplete) reasoners for one of the OWL 2 profiles.

Hence, we consider that the integration of ontologies via mappings seems ideal as reasoning benchmarks.

Acknowledgements

This work was supported by the EU FP7 IP project Optique (no. 318338), the MIUR project CINA (Compositionality, Interaction, Negotiation, Autonomicity for the future ICT society) and the EPSRC project Score!. We also thank the unvaluable help provided by Bernardo Cuenca and Ian Horrocks.

References

1. Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. J. Data Sem. 1, 153–184 (2003)
2. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for OWL. J. Web Sem. 6(4), 309–322 (2008)
3. Cuenca Grau, B., Dragisic, Z., Eckert, K., et al.: Results of the Ontology Alignment Evaluation Initiative 2013. In: Ontology Matching (OM) (2013)
4. David, J., Euzenat, J., Scharffe, F., Trojahn, C.: The Alignment API 4.0. J. Sem. Web 2(1), 3–10 (2011)

5. Dragisic, Z., Eckert, K., Euzenat, J., Faria, D., et al.: Results of the ontology alignment evaluation initiative 2014. In: Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014. pp. 61–104 (2014)
6. Euzenat, J.: Semantic Precision and Recall for Ontology Alignment Evaluation. In: Int'l Joint Conf. on Artif. Intell. (IJCAI). pp. 348–353 (2007)
7. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology Alignment Evaluation Initiative: Six Years of Experience. J. Data Sem. 15, 158–192 (2011)
8. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: Hermit: An OWL 2 reasoner. J. Autom. Reasoning 53(3), 245–269 (2014)
9. Horridge, M.: Justification Based Explanation in Ontologies. Ph.D. thesis, School. Ph.D thesis, School of Computer Science, The University of Manchester (2011)
10. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I.: On the feasibility of using OWL 2 DL reasoners for ontology matching problems. In: OWL Reasoner Evaluation Workshop (2012)
11. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)
12. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences. In: Eur. Sem. Web Conf. (2009)
13. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating Mapping Repair Systems with Large Biomedical Ontologies. In: Description Logics. pp. 246–257 (2013)
14. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. Int'l Sem. Web Conf. (ISWC) pp. 267–280 (2007)
15. Kazakov, Y., Klinov, P.: Goal-directed tracing of inferences in EL ontologies. In: 13th International Semantic Web Conference. pp. 196–211 (2014)
16. Kazakov, Y., Krötzsch, M., Simancik, F.: Concurrent Classification of EL Ontologies. In: Int'l Sem. Web Conf. (ISWC). pp. 305–320 (2011)
17. Meilicke, C.: Alignments Incoherency in Ontology Matching. Ph.D. thesis, University of Mannheim (2011)
18. Meilicke, C., Stuckenschmidt, H.: Incoherence as a basis for measuring the quality of ontology mappings. In: Ontology Matching (OM) (2008)
19. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: Proc. of AAAI Conf. on Artif. Intell. pp. 1408–1413 (2007)
20. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Reasoning Support for Mapping Revision. J. Log. Comput. 19(5) (2009)
21. Pathak, J., Chute, C.G.: Debugging Mappings between Biomedical Ontologies: Preliminary Results from the NCBO BioPortal Mapping Repository. In: Int'l Conf. on Biomedical Ontology (ICBO) (2009)
22. Reiter, R.: A Theory of Diagnosis from First Principles. Artif. Intell. 32(1) (1987)
23. Santos, E., Faria, D., Pesquita, C., Couto, F.: Ontology Alignment Repair Through Modularization and Confidence-based Heuristics. arXiv:1307.5322 preprint (2013)
24. Schlobach, S.: Debugging and Semantic Clarification by Pinpointing. In: Eur. Sem. Web Conf. (ESWC), pp. 226–240. Springer (2005)
25. Schlobach, S., Cornet, R.: Non-standard Reasoning Services for the Debugging of Description Logic Terminologies. In: Int'l Joint Conf. on Artif. Intell. (IJCAI). pp. 355–362 (2003)
26. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowl. and Data Eng. (TKDE) (2012)
27. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Sem. 5(2), 51–53 (2007)
28. Steigmiller, A., Liebig, T., Glimm, B.: Konclude: System description. J. Web Sem. 27, 78–85 (2014)
29. Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A Modularization-Based Approach to Finding All Justifications for OWL DL Entailments. In: Asian Sem. Web Conf. (ASWC) (2008)