WP8-D81

Efficient justification technologies

Project full title:	Knowledge Driven Data Exploitation
Project acronym:	K-Drive
Grant agreement no.:	286348
Project instrument:	EU FP7/Maria-Curie IAPP/PEOPLE WP 2011
Document type:	D (deliverable)
Nature of document:	R (report)
Dissemination level:	PU (public)
Document number:	UNIABDN, IBM/WP8-D81/D/PU/b1
Responsible editors:	Jeff Z. Pan, Jhonatan Garcia, Nophadol Jekjantuk, Yuting Zhao
Reviewers:	Alessandro Faraotti
Contributing participants:	UNIABDN, IBM
Contributing workpackages:	WP8
Contractual date of deliverable:	31 Oct 2015
Actual submission date:	31 Oct 2015

Abstract
Traditionally queries ask for what happens and what does not happen in the knowledge base. While hypothesises take one step forward, tell why something happens and why something does not happen in the knowledge base. In addition to deductive reasoning and explanations of its results. Hypothesis generation requires abductive reasoning support. To support such a new service, the objective of this deliverable is to understand existing abductive reasoning approaches, in which we focus on explaining why certain results can be inferred.

Keyword List
Abductive reasoning, Justification

Project funded by the European Commission within the 7th Framework Programme/Maria Curie Industry-Academia Partnerships and Pathways schema/PEOPLE Work Programme 2011.

Efficient justification technologies

Jeff Z. Pan[1], **Jhonatan Garcia**[1], **Nophadol Jekjantuk**[1], **Yuting Zhao**[2]

[1] Department of Computing Science, Aberdeen University, UK
[2] IBM Italy, Roma, Italy

31 Oct 2015

Abstract
Traditionally queries ask for what happens and what does not happen in the knowledge base. While hypothesises take one step forward, tell why something happens and why something does not happen in the knowledge base. In addition to deductive reasoning and explanations of its results. Hypothesis generation requires abductive reasoning support. To support such a new service, the objective of this deliverable is to understand existing abductive reasoning approaches, in which we focus on explaining why certain results can be inferred.

Keyword List
Abductive reasoning, Justification

Contents

1 Introduction

Workpackage 8 of K-Drive aims to further extend the query generation (WP3, WP5, WP6), to generate not only single queries about "what can be found in the knowledge base", but also "why something happen/not happen in the knowledge base". Actually it applies not only deductive reasoning, but also abductive inferences.

In WP3, we have studied and developed technologies to generate insightful queries from large volume of use case data efficiently and allowing users to conveniently and intuitively browse and exploit data. Technically, it firstly identifies the key dimensions by the experts, or extract key features by automatic extractor, and then generate key queries and compute answers.

In WP5, we have extended the technologies developed in WP3 to generate queries dynamically. This involves the elimination of out-dated queries, and proposal of new queries.

In WP6, we have investigated stream reasoning technologies that can efficiently update query results, in which we focus on be large volume data and its high frequency updating.

Traditionally queries ask for what happens and what does not happen in the knowledge base. While hypothesises take one step forward, tell why something happens and why something does not happen in the knowledge base. In addition to deductive reasoning and explanations of its results. Hypothesis generation requires abductive reasoning support.

Abductive reasoning needs the justifications of computations. Justifications explain why certain results can be inferred. It has been proven that computing all justifications for ontologies in even rather simply knowledge formalisms such as the EL+ used in case of SNOMED CT is quite expensive.

In this deliverable, we focus on technologies for efficient computation of the justifications, especially, how to optimise the computation by exploiting approximate justifications.

In the rest we will introduce the TrOWL ontology reasoning infrastructure and share our experience of using TrOWL to reason with various versions of the Foundational Model of Anatomy Ontology (FMA), which are among the most challenging ontologies for Description Logic reasoners.

We first give a brief introduction of the TrOWL[1] ontology reasoning infrastructure, before we introduce the FMA ontologies (Section 3), some evaluations of using TrOWL to reason with the FMA ontologies (Sections 4 and 5).

2 TrOWL Ontology Reasoning Infrastructure

TrOWL[2] is a tractable reasoning infrastructure for the second version of the Web Ontology Language, or simply OWL2[3], which comes with a family of ontology languages, including:

- OWL2-DL, the most expressive decidable language in the OWL2 family, and
- three tractable sub-languages of OWL2-DL, i.e. OWL2-EL, OWL2-QL and OWL2-RL.

There are at least three approaches to reasoning in OWL2:

1. Sound and complete reasoning in OWL2-DL. Until recently, no OWL2-DL reasoners could classify the FMA ontology, due to the high worst case complexity of OWL2-DL (2NEXP-TIME-complete). In 2010, Glimm et. al. Glimm et al. (2010a) proposed the core blocking optimisation,

[1] http://trowl.eu/
[2] http://trowl.eu/
[3] http://www.w3.org/TR/owl2-overview/

enabling HermiT[4] to classify the TBox of the FMA-Constitutional ontology in about 30 minutes Golbreich et al. (2011).

2. Sound and complete reasoning in OWL2-EL, OWL2-QL and OWL2-RL. Although these sub-languages are tractable, none of them are sufficiently expressive to cover FMA.

3. Approximate reasoning for OWL2-DL. The idea here is to approximate OWL2-DL ontologies to those in its tractable sub-languages, so as to exploit the efficient and scalable reasoner. In Sections 3 and 4 of the paper, we will provide more details on the performance of our approximate reasoner in TrOWL for the FMA ontologies.

TrOWL supports OWL2 by using the approaches 2 and 3 mentioned above. On the tractable language level, TrOWL contains an OWL2-EL reasoner (REL) and an OWL2-QL reasoner (Quill). The approach of TrOWL is to offer tractable support for all the expressive power of OWL2 by using quality guaranteed (in terms of soundness and/or completeness) approximate reasoning. TrOWL contains an OWL2 profile checker to detect which profile an ontology may fit into.

The semantic approximation from OWL2 to OWL-QL is based on the work described in Pan and Thomas Pan and Thomas (2007). Semantic approximation applies the knowledge compilation Selman and Kautz (1996) to precompute the entailment of an arbitrary ontology into a DL-Lite ontology. In general, this approach is soundness preserving and could be incomplete. Furthermore, a conditional completeness condition is identified: if input queries are database style queries, i.e. the variables only bound to named individuals, the approach is also complete. In other words, database style queries are not expressive enough to tell the difference between the original OWL2-DL ontology and its approximation. A drawback for this approach is that reasoners are required to compute the semantic approximation; therefore, the construction of the approximation is usually done off-line.

The syntactic approximation from OWL2 to OWL2-EL is based on the soundness preserving approximate reasoning approach presented in Ren et al. Ren et al. (2010). The construction of the approximation is on the syntax and hence can be done efficiently just before applying approximate reasoning. The idea is not to throw away the axioms that are beyond OWL2-EL; otherwise, we might suffer from low recall — for example, if we naively remove from the Cyc ontology all the axioms that are beyond OWL2-EL, the recall is only 1% for classification. Therefore, in this approach, we introduce some fresh named classes to represent non-OWL2-EL class expressions. In order to recover the hidden semantics within these fresh named classes, some relation between such named classes and existing classes are maintained and some extra completion rules (beyond those in OWL2-EL) are introduced, with the extended set of completion rules still being tractable. In Ren et al. (2010), we reported that the recall of such approximate reasoning is very high for TBox classification, 100% for most existing benchmark ontologies except the Wine Ontology (99.4%). A further investigation indicates that it is due to the syntax sensitivity nature of our approach. After adding a further normalisation step into TrOWL, the recall for the Wine Ontology is also 100%.

TrOWL supports both OWL and Jena APIs. It has a plug-in for the Prot$\acute{e}g\acute{e}$ ontology editor v4.3.

3 FMA Ontologies

The Foundational Model of Anatomy Ontology (FMA) Rosse and Jr. (2003) is an evolving comput-er-based knowledge source for biomedical informatics, mainly developed by the University of Washington since 1994. The importance of this ontology resides in the fundamental underlining of anatomy in all

[4] http://www.hermit-reasoner.com/

File	Source	File size	Cls	Props	Inds	Expressivity
FMA-DLR[4]	bioontology.org	147.6 Mb	78989	110	139374	$\mathcal{ALUIN}(\mathcal{D})$
FMA-FullR[4]	bioontology.org	37.7 Mb	23597	77	82935	$\mathcal{ALCOF}(\mathcal{D})$
FMA-Constitutional	Golbreich et al. (2006)	42.1 Mb	41647	148	85	$\mathcal{ALCOIF}(\mathcal{D})$
FMA-OWL2G_noMTC	Golbreich et al. (2011)	261.7 Mb	85005	140	74698	$\mathcal{SROIQ}(\mathcal{D})$
FMA-DLR_M1[5]	this work	140Kb	26	133	26	$\mathcal{ALCOIF}(\mathcal{D})$
FMA-DLR_M2[6]	this work	225Kb	56	133	56	$\mathcal{ALCOIF}(\mathcal{D})$

Table 1: List of FMA ontologies

fields of medicine. Proper interpretation of these data relies on an implicit understanding of anatomy. The inferences entailed in such reasoning call upon cognitive or computational processing of abstractions about physical entities of the body, making use of relationships that exist among anatomical concepts. Relevance and impact of the FMA ontology on its field can now be compared to other well-known medical ontologies such as SNOMED Q. et al. (2001) or GALEN JE et al. (2001).

Nowadays, the FMA can be viewed as a complex, highly connected network in which nearly 70,000 anatomical concepts, from over 170,000 frames, are interrelated by over 570,000 relationship instances. There have been a number of approaches Golbreich et al. (2006); Noy and Rubin (2008); Golbreich et al. (2011) to translating the knowledge encapsulated by the FMA ontology into the OWL ontology language. Golbreich et al. Golbreich et al. (2011) developed the FMA-OWLizer tool, which can be applied to automatically obtain a translation of the FMA ontology into OWL 2.

Given the size and complexity of the FMA ontology, reasoning under OWL has proven to be a real challenge. Table 1 provides a list of FMA ontologies written in OWL that we used in our evaluation.

4 FMA with Metamodeling

The FMA features a complex structure of superclasses and subclasses that requires the support of metamodeling. For example, "Physical anatomical entity" is an instance of "Anatomical entity template", and a subclass of both "Anatomical entity template" and "Anatomical entity" Dameron et al. (2005). In OWL, a class is interpreted as a set of objects. Similarly, a metaclass is interpreted as a set of sets in metamodeling extensions of OWL, such as OWL-FA Jekjantuk et al. (2008). For example, the metaclass Vertebra can be interpreted as a set of different types of vertebrae, such as cervical, thoracic, lumbar, which in turn can be interpreted as subsets of other sets, e.g., first, ..., fifth lumbar vertebra.

There have been several attempts in dealing with metamodeling in FMA. Dameron et al. Dameron et al. (2005) converted the frame-based FMA ontology into an OWL1-DL version and an OWL1-Full version, with metaclasses included in the latter one. Golbreich et al. Golbreich et al. (2006) tried to capture (some of) the knowledge encoded at metaclasses differently in OWL1-DL directly (cf. the FMA-Constitutional ontology in Table 1). The idea is to replace instance-of links between a class and its metaclasses with subClassOf links. The structure of their instances, property restrictions of metaclasses are interpreted as closure axioms and approximated by universal restrictions while restrictions of classes are translated into existential restriction. Later on, Golbreich et al. further encoded the FMA ontology in OWL2-DL, producing two ontologies, one with metaclasses and one without them. The

[4] http://www.bioontology.org/wiki/index.php/FMAInOwl
[5] http://homepages.abdn.ac.uk/jeff.z.pan/pages/onto/fma-dlr-m1.owl
[6] http://homepages.abdn.ac.uk/jeff.z.pan/pages/onto/fma-dlr-m2.owl

idea is to use the OWL 2 metamodeling capability, i.e., punning, to represent metaclasses, using the same URI to refer to a class and an individual at the same time in FMA-OWL2G_MT. For example, the name `Heart` can be used both for the metaclass `Heart` and for the class `Heart`, instance of `Organ_with_cavitated_organ_parts`.

The drawback of the punning approach is that, although a class and an individual can share the same name, say C, they are treated as different entities. For example, even if the class C is entailed to be equivalent to a class D, the individuals C and D can still be different. This has been regarded as non-intuitive due to the lack of expected entailments (e.g., the individuals C and D should be the same). To deal with this problem, we apply the class-based approach from Glimm et al. Glimm et al. (2010b) to enrich some small (but already challenging for DL reasoners) subsets of FMA-DLR ontology and accommodate metaclasses (cf. the last two ontologies in Table 1) with the `Typing` and `MatSubClass` functions proposed in Glimm et al. (2010b).

Ontology	RT	FaCT++	HermiT	TrOWL	MORe	Recall
FMA-DLR	C	32.425 s	46.94 s	32.596 s	47.794 s	100%
FMA-FullR	C	121.041 s	1064.947 s	4.45 s	4.571 s	100%
FMA-Constitutional	C	t/o	3043.61 s	155.808 s	t/o	100%
FMA-OWL2G_noMTC	C	o/m	t/o	967.59 s	t/o	N/A
FMA-DLR_M1	M	932.5 s	26.39 s	0.819 s	N/A	100%
FMA-DLR_M2	M	t/o	737.43 s	2.863 s	N/A	100%

Table 2: Reasoning with FMA ontologies via OWL API (‘RT’ for Reasoning Task, ‘C’ for Classification, ‘M’ for Materialisation, ‘s’ for second, ‘t/o’ for time out after one hour, ‘o/m’ for out of memory)

Table 2 lists the classification (for the first four ontologies) and materialisation (for the last two ontologies) time (reasoning time + retrieving time) from some of the state of the art DL reasoners (including FaCT++ v1.6.2, HermiT v1.3.8, TrOWL v1.3 and MORe v0.1.3 Romero et al. (2012)) over the FMA ontologies in Table 1. The machine used for the experiment is a MacBook, with CPU 2.26 core 2 duo, Ram 8 GB and 6 GB allocated to JVM. The last column of Table 2 reports the recall (on the number of subsumptions among named classes) of TrOWL with respect to results of HermiT. Note that FaCT++ had a datatype error for the FMA-DLR ontology, so we did not import the FMA-DLR ontology into the FMA-FullR ontology when testing FMA-FullR. Moreover, as MORe does not support ABox reasoning, its time with the FMA-DLR_M1 and FMA-DLR_M2 ontologies is not reported.

5 Dealing with Unsatisfiable Concepts in FMA Constitutional

An ontology is called *incoherent* Schlobach and Cornet (2003) if it contains unsatisfiable concepts, which are equivalent to the bottom concept $\bot$ and can not have any instance. Unsatisfiable named concepts (except the bottom concept $\bot$) in a constructed ontology usually indicates possible design flaws. For example, 33,433 out of 41,648 concepts are unsatisfiable in the FMA-Constitutional ontology. This was apparently not intended since the current version of FMA has already eliminated all these unsatisfiabilities.

Understanding ontology incoherence is not trivial. Incoherence can usually be explained and resolved by computing justifications Kalyanpur et al. (2007), i.e., minimal entailment-preserving sub-ontologies. However, computing justifications with a black-box algorithm requires a large number of entailment checking, which can be expensive given the complexity of reasoning and size of the ontology. Also, looking into 33,433 justifications to debug the ontology will be very time consuming.

We notice that some of the concepts are unsatisfiable due to other unsatisfiable concepts. For example in FMA-Constitutional we can infer $Neuron \sqsubseteq \bot$ and $Central\_neuron \sqsubseteq Neuron$, hence we also have $Central\_neuron \sqsubseteq \bot$. In this case, $Central\_neuron$ is unsatisfiable due to the unsatisfiability of $Neuron$. Such a phenomena has been formally characterised by Kalyanpur et al. Kalyanpur et al. (2005) as *root* and *derived* unsatisfiable concepts. Particularly, A is a derived unsatisfiable concept if there is a justification for $A \sqsubseteq \bot$ that contains a justification for $B \sqsubseteq \bot$, where B is another unsatisfiable concept, and B is called the *parent* of A. Otherwise, A is a *root* unsatisfiable concept.

The REL reasoner in TrOWL is using a forward-chaining completion-based algorithm, in which each rule infers a set of consequence axioms from a set of antecedence axioms. Such an algorithm can be easily extended to compute justifications on the fly by incorporating a truth-maintenance system (TMS) Ren and Pan (2011). However naively applying such a solution has the following limitations:

1. For big and complex ontologies, maintaining the entire in TMS is a big overhead on reasoning. In fact, we are only interested in justifications for unsatisfiability but not the others so a fully-fledged TMS is unnecessary.

2. Repairing all the *root* unsatisfiabilities cannot always repair all the *derived* unsatisfiabilities because a derived unsatisfiability may have another justification that depends on no other unsatisfiability. In this case, one need to iteratively reclassify, debug and repair. For difficult ontologies such as FMA Constitutional, we would like to minimise the number of such iterations.

In order to improve efficiency, we introduce Type I and Type II unsatisfiable concepts as approximations to the root and derived unsatisfiable concepts with the following procedure in REL:

1. When a rule infers $A \sqsubseteq \bot$, if the antecedences contain $B \sqsubseteq \bot$, where both A and B are named concepts in the original ontology, we label A as a *Type II* unsatisfiable concept. Otherwise, we label A as a *Type I* unsatisfiable concept.

2. We continue reasoning on Type II concepts regardless their unsatisfiability, and label them with Type I if possible. They will not be treated immediately as sub-concept of all concepts.

The above point 2 is important to explore alternative derivation of unsatisfiability for Type II concepts. If such a derivation does not depend on other unsatisfiability, the concept will be labeled Type I as well. It is possible for a derived unsatisfiable concept to be labeled earlier than its parent, making it mistakenly labeled as a Type I. For example, considering the following axiom:

$$A \sqsubseteq B, \tag{1}$$

$$B \sqsubseteq C, \tag{2}$$

$$B \sqsubseteq \neg C, \tag{3}$$

if we infer $A \sqsubseteq C$ from (1) and (2), then $A \sqsubseteq \neg C$ from (1) and (3), and then $A \sqsubseteq \bot$, then we have A as a Type I unsatisfiable concept. To avoid such situations as much as possible, we apply a depth-first classification strategy, always classifying super-concepts before classifying sub-concepts.

Using the above mechanism we are able to distinguish the different types of unsatisfiable concepts in the FMA-Constitutional. Results show that only 145 concepts belong to Type I, which is only 0.43% of all the unsatisfiable concepts. By examining their justifications, we realise that they are due to similar reasons. Particularly, there is a boolean-valued functional datatype property has_mass in the ontology. With the axioms in FMA Constitutional, it is possible to infer both $A \sqsubseteq \exists has\_mass.\{true\}$ and

$A \sqsubseteq \exists has\_mass.\{false\}$ for concept A, making A unsatisfiable. Another boolean-valued functional datatype property $has\_inherent\_3\text{-}D\_shape$ has a similar problem. There are in total only 6 concept axioms with these two properties. Debugging these 6 axioms is apparently much easier than debugging all the 122,136 logical axioms, or the justifications of all the 33,433 unsatisfiable concepts.

6 Conclusion

The objective of Workpackage 8 is to generate hypothesis, which, if presented by queries we do in the K-Drive project, is about "why something happen/not happen in the knowledge base". Traditionally queries ask for what happens and what does not happen in the knowledge base. While hypothesises take one step forward, tell why something happens and why something does not happen in the knowledge base. In addition to deductive reasoning and explanations of its results.

Hypothesis generation requires abductive reasoning support, while abductive reasoning needs the justifications of computations. Justifications explain why certain results can be inferred. It has been proven that computing all justifications for ontologies in even rather simply knowledge formalisms such as the EL+ used in case of SNOMED CT is quite expensive.

Our focus in this deliverable is on the technologies for efficient computation of the justifications. Actually we have introduced the TrOWL[5] ontology reasoning infrastructure, and shown the evaluations with the FMA ontologies, which indicates that approximate reasoners can be useful for reasoning and debugging complex ontologies.

Acknowledgement

This research has been funded by the European Commission within the 7th Framework Programme/Marie Curie Industry-Academia Partnerships and Pathways schema/PEOPLE Work Programme 2011 project K-Drive number 286348 (cf. `http://www.kdrive-project.eu`).

References

Dameron, O., Rubin, D. L., and Musen, M. A. (2005). Challenges in converting frame-based ontology into OWL: the Foundational Model of Anatomy case-study. In *the Proc. of the AMIA Annual Symposium*.

Glimm, B., Horrocks, I., and Motik, B. (2010a). Optimized description logic reasoning via core blocking. In *IJCAR*, pages 457–471.

Glimm, B., Rudolph, S., and Völker, J. (2010b). Integrated metamodeling and diagnosis in owl 2. In *Proceedings of the 9th International Semantic Web Conference*, volume 6496 of *LNCS*, pages 257–272. Springer.

Golbreich, C., Grosjean, J., and Darmoni, S. J. (2011). The fma in owl 2. In *Proceedings of the 13th conference on Artificial intelligence in medicine*, AIME'11, pages 204–214, Berlin, Heidelberg. Springer-Verlag.

[5]http://trowl.eu/

Golbreich, C., Zhang, S., and Bodenreider, O. (2006). The foundational model of anatomy in owl: Experience and perspectives. *J. Web Sem.*, 4(3):181–195.

JE, R., A, R., WD, S., van der Haring E, CJ, W., PE, Z., and Rector, A. (2001). Galen ten years on: Tasks and supporting tools. In *Proceedings of the MEDINFO2001*, pages 256–260. IOS Press.

Jekjantuk, N., Pan, J. Z., and Groner, G. (2008). Verifying and Validating Multi-Layered Models with OWL FA Toolkit. In *the Proc. of the Extended Semantic Web Conference (ESWC2010).*

Kalyanpur, A., Parsia, B., Horridge, M., and Sirin, E. (2007). Finding all justifications of owl dl entailments. In *ISWC/ASWC 2007*, pages 267–280.

Kalyanpur, A., Parsia, B., Sirin, E., and Hendler, J. (2005). Debugging unsatisfiable classes in owl ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(4):268–293.

Noy, N. F. and Rubin, D. L. (2008). Translating the foundational model of anatomy into owl. *J. Web Sem.*, 6(2):133–136.

Pan, J. Z. and Thomas, E. (2007). Approximating OWL-DL Ontologies. In *the Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1434–1439.

Q., S. M., C., P., A., S. K., and Y., W. A. (2001). Snomed clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium, American Medical Informatics Association*, pages 662–666.

Ren, Y. and Pan, J. Z. (2011). Optimising ontology stream reasoning with truth maintenance system. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 831–836. ACM.

Ren, Y., Pan, J. Z., and Zhao, Y. (2010). Soundness Preserving Approximation for TBox Reasoning. In *the Proc. of the 25th AAAI Conference Conference (AAAI2010).*

Romero, A. A., Grau, B. C., and Horrocks, I. (2012). More: Modular combination of owl reasoners for ontology classification. In *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, LNCS. Springer.

Rosse, C. and Jr., J. L. V. M. (2003). A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, 36(6):478–500.

Schlobach, S. and Cornet, R. (2003). Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI*, pages 355–362. Morgan Kaufmann.

Selman, B. and Kautz, H. (1996). Knowledge compilation and theory approximation. *J. ACM*, 43(2):193–224.