2013 IEEE 25th International Conference on Tools with Artificial Intelligence

A Novel Combination of Reasoners for Ontology Classification

Changlong Wang[1,2,3], Zhiyong Feng[1,2]
[1]*School of Computer Science and Technology, Tianjin University,Tianjin 300073, China*
[2]*Tianjin Key Laboratory of Cognitive Computing and Application,Tianjin 300073, China*
[3]*School of Computer Science and Engineering,Northwest Normal University,Lanzhou 730070, China*
Email: wcanglong@gmail.com,zyfeng@tju.edu.cn

***Abstract*—Large scale ontology applications require efficient reasoning services, of which ontology classification is the fundamental reasoning task. The special EL reasoners are efficient, but they can not classify ontologies with axioms outside the OWL 2 EL profile. The general-purpose OWL 2 reasoners for expressive Description Logics are less efficient when classifying the OWL 2 EL ontologies. In this work, we propose a novel technique that combines an OWL 2 reasoner with an EL reasoner for classification of ontologies expressed in DL SROIQ. We develop an efficient task decomposition algorithm for identifying the minimal non-EL module that is assigned to the OWL 2 reasoner, and the bulk of the workload is assigned to the EL reasoner. Furthermore, this paper reports on the implementation of our approach in the ComR system which integrates the two types of reasoners in a black-box manner. The experimental results show that our method leads to a reasonable task assignment and can offer a substantial speedup (over 50%) in ontology classification.**

***Keywords*-description logic; OWL 2; OWL 2 EL; reasoner; combination; ontology classification**

I. INTRODUCTION

Ontologies expressed in the Web Ontology Language (OWL) and the second version OWL 2 play a central role in the development of the Semantic Web [1]. They are also widely used in biomedical information systems and other area [2], [3], [4]. Those systems based on ontology require efficient and robust reasoning services. The fundamental reasoning task is ontology classification, which determines subconcept／supconcept relationship (called subsumption relationship) between the concepts of a given terminology, and thus allows one to construct the terminology in the form of subsumption hierarchy. This hierarchy provides useful information on the connection between different concepts and can be used to speed-up other inference service. The classification results are also useful for navigating the ontology and identifying modeling errors, as well as for explanation and query answering.

It is well known that the expressive Description Logics (DLs) have high worst case computational complexity. For example, classification in the DL SROIQ, the underlying logic of OWL 2, is N2EXPTIME-complete [5]. In spite of the theoretical complexity result, some highly optimised reasoners such as hermiT [6], RacerPro [7], Pellet [8], and FaCT++ [9] are able to classify many ontologies in applications. Most of these reasoners are based (Hyper)Tableau algorithm and build models to check concept satisfiability. Moreover, those general-purpose reasoners provide some dedicated optimizations for ontology classification. FaCT++ reduces the number of subsumption tests for completely defined concepts, an extension of this optimization with structural pseudo-model embedding has been used by RacerPro. HermiT uses an optimization that can completely avoid subsumption tests for deterministic ontologies—subsumptions can be just read out of the models produced for concept satisfiability tests. The latest version of Pellet can apparently switch to a specialized procedure when the ontology is within an OWL 2 EL profile [10].

Although using many optimizations, those general-purpose OWL 2 reasoners mentioned above still take long time to classify large ontologies such as Protein, OBI and ICNP [11]. This has stimulated researchers to pay more attention to another tractable ontology language——OWL 2 EL, one of OWL 2 profiles, based on the lightweight description logic EL^{++}, for which most standard reasoning tasks can be performed in polynomial time [12]. Several very efficient profile-specific reasoners have been developed for OWL 2 EL, including CEL [13], jcel [14], and ELK [15]. These reasoners are based on specifically designed inference rules for deriving logical consequences of the axioms in ontology, and they are significantly faster than OWL 2 reasoner. Unfortunately, an EL reasoner is only able to completely classify ontologies expressed in OWL 2 EL fragment.

Many commonly used ontologies, such as Open Biomedical Ontologies (OBO) [16], are covered by OWL 2 EL to a large degree. For example, of the 219,224 axioms in the later version of the National Cancer Institute Ontology (NCI), only 65 are outside the OWL 2 EL fragment, and of the 46698 axioms in the Protein ontology, only 16 are outside the OWL 2 EL fragment. This motivate us to take full advantages of exiting OWL 2 reasoner and EL reasoner—combining the two types of reasoners to classify those ontologies expressed in SROIQ language.

In this paper, we propose a novel technique which combines a reasoner for some more expressive DLs (e.g. SROIQ) and another one for restricted DLs (e.g. EL^{++}). A distinguishing property of our technique is that it can take

1082-3409/13 $31.00 © 2013 IEEE
DOI 10.1109/ICTAI.2013.75

full advantages of the two type of reasoners—the bulk of the workload is assigned to the efficient EL reasoner (called Main Reasoner, MR), and the minimal non-EL module is given to OWL 2 reasoner (called Assistant Reasoner, AR). More precisely, given an OWL 2 ontology O, the classification procedure proceeds as follows:

1. Task decomposition — computing the non-EL module $O_{\overline{EL}} \subseteq O$ such those classes in the $O_{\overline{EL}}$ can be completely classified using only the axioms in the $O_{\overline{EL}}$.

2. Partial classification — using AR to classify the $O_{\overline{EL}}$, the classification result CR($O_{\overline{EL}}$) falls into the OWL 2 EL fragment.

3. Constructing a temporary ontology — using all the axioms in CR($O_{\overline{EL}}$) and $(O \setminus O_{\overline{EL}})$ to construct a temporary ontology $O^{'}$.

4. Entire classification — using MR to classify the ontology $O^{'}$.

There are two important technical challenges in the step 1. First, $O_{\overline{EL}}$ should be as small as possible, in particular, for ontologies with only a few non-EL axioms, it is reasonable to expect to contain only the non-EL axioms. Second, $O_{\overline{EL}}$ must be complete for reasoning, i.e., for two classes *A*, *B* in $O_{\overline{EL}}$, $O_{\overline{EL}} \models A \sqsubseteq B$ iff $O \models A \sqsubseteq B$. Although module extraction technique can be used directly to identify a fragment of an ontology that is complete for a certain signature, this technique is more complicated. We exploit the ontology decomposition technique [17], which allow us to assembly efficiently modules before reasoning and complete task decomposition off-line.

The remainder of this paper is structured as follows. After a brief introduction to the preliminaries, in section III we study ontology decomposition technique and develop an algorithm for building the minimal non-EL module $O_{\overline{EL}}$. In section IV, we give an algorithm for ontology classification in the frame of combined reasoners and describe the basic workflow. In section V, we reports on the implementation, experiments, and discussion. In section VI, we draw some conclusions and outline issues for future work.

II. PRELIMINARIES

We assume the reader to be reasonably familiar with OWL 2 and its profile OWL 2 EL [18], which have became the W3C standard.

For convenience reasons, we adopt description logic notion rather than OWL syntax in the definitions and other formal descriptions, hence we assume the readers are familiar with the syntax and the semantics of the DLs SROIQ [19] and EL^{++} [12], which are the logical basics of the OWL 2 and OWL 2 EL, respectively.

We denote OWL 2 ontology with O, and OWL 2 EL ontology with O_{EL}, which only contains EL axioms. We use $O_{\overline{EL}}$ to denote an ontology containg axioms in $O \setminus O_{EL}$. With $\widetilde{\alpha}$ we denote the signatures in the axiom α.

We use the notion of a locality-based module [20], which is a subset of an ontology O and preserves all consequences of O w.r.t. a signature Σ. Given an ontology O, a seed signature Σ, and a module notion $x \in (\top, \bot, \top\bot^*)$, we denote the x-module w.r.t. Σ with x-mod(Σ,O).

III. TASK DECOMPOSITION

The goal of task decomposition is to obtain a subontology $O_{\overline{EL}}$ from the original ontology O, such that $O_{\overline{EL}}$ must contains all the non-EL axioms in O and should be as small as possible. We exploit the technique of ontology decomposition [17], which provide an efficient approach to build module.

A. ATOMIC DECOMPOSITION OF ONTOLOGY

Atomic decomposition (AD) is an approach to represent the whole family of locality-based *x*-modules of an ontology O. The key point is observing that some axioms appear in a module only if other axioms do. This allow us to describe the logical dependence between axioms.

Definition 3.1 Given an axiom α in ontology O, the x-mod($\widetilde{\alpha}$,O) is called α-module, denoted with M_α.

Remark 3.2 [21] For each axiom $\alpha \in O$, the module M_α is the smallest x-module containing α .

Definition 3.3 Let α and β be two distinct axioms of an ontology O, α depends on β if $M_\beta \subseteq M_\alpha$.

The dependence between axioms allows us to identify clumps of highly interrelated axioms that are never split across two or more modules; these clumps are called *atoms*. More precisely, for $x \in (\top, \bot, \top\bot^*)$, an *x*-atom is a maximal subset of an ontology O which is either contained in, or disjoint with any *x*-module of O. The family of *x*-atoms of O is called *x*-atomic decomposition (*x*-AD). If *x* is clear from the context, we drop it. Given an ontology O, the AD(O) can be computed via α-module only.

The algorithm for ontologies atomic decomposition [17] is given in Algorithm 1. Given ontology O, this algorithm computes the AD(O) w.r.t the type of *x*-module.

Since every atom is a set of axioms, and atoms are pairwise disjoint, Hence, the AD(O) is a partition of the ontology O. Moreover, the atoms are the building blocks of all modules [17]. For an atom $\mathbf{a} \in$ AD(O), the module $M_{\mathbf{a}}$=*x*-mud($\widetilde{\mathbf{a}}, O$) is called *compact*.

Proposition 3.4 Given an ontology O, Let atom $\mathbf{a} \in$ AD(O) and axiom $\alpha \in \mathbf{a}$, for any selection axioms S=$\{\alpha_1, \ldots, \alpha_m\} \subseteq \mathbf{a}$, x-mod($\widetilde{S}$,O)=M_α. In particular, for each $\alpha_i \in \mathbf{a}$, M_{α_i}=M_α. Vice versa, if M_α=M_β, then there exists some atom $\mathbf{a}$ such that $\alpha, \beta \in \mathbf{a}$.

As a consequence of Proposition 3.4, the set of compact modules coincides with the set of α-modules. With M_α we denote the module $M_{\mathbf{a}}$ for $\alpha \in \mathbf{a}$, then the definition of logical dependence can be extended to atoms.

Definition 3.5 Let $\mathbf{a}$ and $\mathbf{b}$ be two distinct atoms of an ontology O, $\mathbf{a}$ is dependent on $\mathbf{b}$ (written $\mathbf{a} \succeq \mathbf{b}$) if $M_{\mathbf{b}} \subseteq M_{\mathbf{a}}$.

Algorithm 1 Atomic Decomposition

```
Input: An ontology O
Output: the poset of atoms (AD(O), ⪰)
TodoAxioms ← x-mod(Õ, O)\ x-mod(∅, O)
GenAxioms ← ∅
for each α ∈ TodoAxioms do
   Module(α) ← x-mod(α̃, O)
   new ← true
   for each β ∈ GenAxioms do
      if Module(α)=Module(β) then
         Atom(β)← Atom(β)⋃{α}
         new ← false
      end if
   end for
   if new=true then
      Atom(α)← {α}
      GenAxioms ← GenAxioms⋃{α}
   end if
end for
for each α ∈ GenAxioms do
   for each β ∈ GenAxioms do
      if β ∈Module(α) then
         Atom(β) ⪰ Atom(α)
      end if
   end for
end for
AD(O)← {Atom(α) | α ∈ GenAxioms }
return (AD(O), ⪰)
```

Proposition 3.6 Given an ontology O, the dependency relation $\succeq$ is a partial order over the AD(O).

Definition 3.5 and proposition 3.6 allow us to draw a Hasse diagram to present the AD(O), where nodes are atoms and edges present the dependency relationship between two atoms. For two atoms **a** and **b**, if **a** $\succeq$ **b**, there is no atom **c** distinct from **a** and **b** such that **a** $\succeq$ **c** $\succeq$ **b**; we draw an edge from **a** to **b**, and **a** is in the position higher than **b**'s. See Figure 1, the Hasse diagram for Example 1 ontology.

Example 1. Consider the ontology O=$\{\alpha_1, \cdots, \alpha_7\}$ and its $\bot$-AD:

α_1: Animal $\sqsubseteq$ (= 1hasGender.$\top$);
α_2: Animal $\sqsubseteq$ ($\geq$ 1hasHabitat.$\top$);
α_3: Person $\sqsubseteq$ Animal;
α_4: Vegan $\equiv$ Person $\sqcap$ $\forall$eats.(Vegetable $\sqcup$ Mushroom);
α_5: TeeTotaller $\equiv$ Person $\sqcap$ $\forall$drinks.NonAlcoholicThing;
α_6: Student $\sqsubseteq$ Person $\sqcap$ $\exists$hasHabitat.University;
α_7: GraduateStudent $\equiv$ Student $\sqcap$ $\exists$hasDegree.{BA, BS};

Using the $\bot$-AD, we get 6 atoms: $\mathbf{a}_1 = \{\alpha_1, \alpha_2\}$, $\mathbf{a}_2 = \{\alpha_3\}$, $\mathbf{a}_3 = \{\alpha_4\}$, $\mathbf{a}_4 = \{\alpha_5\}$, $\mathbf{a}_5 = \{\alpha_6\}$, $\mathbf{a}_6 = \{\alpha_7\}$.

B. BUILDING THE MINIMAL $O_{\overline{EL}}$

Intuitively, we can easily get all compact modules of an ontology O from the Hasse diagram for the AD(O).

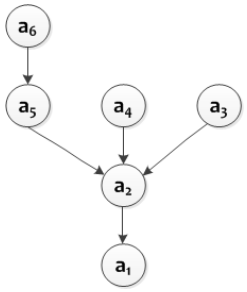

Figure 1. The Hasse Diagram of Example ontology

Definition 3.7 Let atom **a** $\in$ AD(O), The principal ideal of **a** is the set (**a**]= $\{\alpha \in \mathbf{b} \mid \mathbf{a} \succeq \mathbf{b}\}$.

Proposition 3.8 For every atom **a** $\in$ AD(O), (**a**] is a module.

Proposition 3.8 presents the relationship between modules and atoms. Given an ontology O and its AD(O), we can efficiently assemble intended modules related certain atoms. For example, the compact module for the atom $\mathbf{a}_5$ is $M_{\mathbf{a}_5} = \{\mathbf{a}_5, \mathbf{a}_2, \mathbf{a}_1\}$.

Dentition 3.9 For any atom **a** $\in$ AD(O), if the atom **a** contains an axiom that is outside OWL 2 EL fragment, **a** is called non-EL atom.

Once we identify a non-EL atom **a**, we can obtain the non-EL compact module $M_{\mathbf{a}}$ from the dependent chain in the Hasse diagram. Furthermore, we can compute all the non-EL compact mudules of an ontology O, which compose the subontology $O_{\overline{EL}}$.

Proposition 3.10 Let S be the set that contains all the non-EL atoms in AD(O) of an ontology O, and for any **a** $\in AD(O) \bigcap S$, **a** is a non-EL atom, then

$$O_{\overline{EL}} = (\mathbf{a}_1] \bigcup, \cdots, \bigcup(\mathbf{a}_k] \text{ , (} \mathbf{a}_i \in S, i = 1, \cdots, k \text{).}$$

Obviously, $O_{\overline{EL}}$ is a set of compact modules and it is also a module of O, for any axiom $\alpha \in O_{\overline{EL}}$, the following condition holds: $O_{\overline{EL}} \models \alpha$ iff $O \models \alpha$.

Remark 3.11 Given AD(O), $O_{\overline{EL}}$ is the minimal module containing all non-EL atoms in AD(O).

Proof. Given ontology O, let atom **a** $\in AD(O)$, axiom $\alpha \in$ **a**, we have that the principal of **a**, M_α, and $M_{\mathbf{a}}$ coincide [15, Lemma 4.2]. Moreover, M_α=x-mod($\widetilde{\alpha}, O$) is the the smallest module containing **a** [15, Remark 3.10]. Hence, for an atom **a**, its principal is the smallest module containing itself, and $O_{\overline{EL}}$ is the minimal module containing all non-EL atoms in AD(O). □

Algorithm 2 sketches our algorithm for computing $O_{\overline{EL}}$ that runs in polynomial time in the size of AD(O).

IV. CLASSIFICATION WITH THE COMBINED REASONERS

After obtaining the minimal non-EL module $O_{\overline{EL}}$, next we can proceed to classify an ontology O with our combined reasoners (ComR) system, where OWL 2 reasoner and EL

Algorithm 2 getNonELModule(O)

Input: An ontology O
Output: $O_{\overline{EL}}$
$O_{\overline{EL}} \leftarrow \emptyset$
$AD(O) \leftarrow getAD(O)$ //Algorithm 1
for each **a** $\in AD(O)$ **do**
  if a is non-EL atom **then**
    $O_{\overline{EL}} \leftarrow O_{\overline{EL}} \bigcup$[**a**]
  end if
end for
return $O_{\overline{EL}}$

reasoner are combined to classify an OWL 2 ontology. As already mentioned in the previous sections, we can classify the non-EL module $O_{\overline{EL}}$ using only OWL 2 reasoner (acting as AR) and get the classification result CR($O_{\overline{EL}}$), a set of simple atomic subsumption axioms of the form $A \sqsubseteq B$, which falls into the OWL 2 EL fragment. Finally, we construct a temporary ontology $O^{'}$ containing axioms in CR($O_{\overline{EL}}$) and $O \backslash O_{\overline{EL}}$. Obviously, $O^{'} \models O$ and $O^{'}$ is an EL ontology which can be assigned to the EL reasoner(acting as MR) for a complete classification for original O. Algorithm 3 describes the entire classification process and Figure 2 demonstrates the basic frame and workflow of the ComR system, where the TD (Task Decompostion) component is to compute the minimal non-EL module $O_{\overline{EL}}$.

Algorithm 3 classifyWithComR(O)

Input: an OWL 2 ontology O
Output: The classification result R
$O_{\overline{EL}} \leftarrow$ getNonELModule(O) //Algorithm 2
CR($O_{\overline{EL}}$) $\leftarrow$ AR.classify ($O_{\overline{EL}}$)
// using OWL 2 reasoner
constructing a temporary ontology $O^{'}$
//using axioms in CR($O_{\overline{EL}}$) $\bigcup (O \setminus O_{\overline{EL}})$
R$\leftarrow$ MR.classify($O^{'}$)
// using EL reasoner
return R

V. EXPERIMENTS AND DISCUSSION

A. EXPERIMENTS

We have implemented the ComR system in Java using the OWL API [23]. The implementation of computing non-

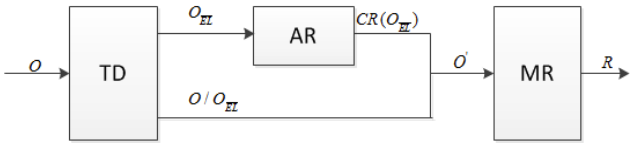

Figure 2. The Fram of ComR

Table I
RESULT OF TASK DECOMPOSITION

Ontology	#O	#NonELAxiom	#NonELAtom	$\#O_{\overline{EL}}$
Protein	46698	16	16	288(0.62%)
NCI	219224	65	57	7151(3.26%)
ICNP	11901	38	35	682(5.73%)
OBI	28492	358	228	5558 (19.51%)

Table II
CLSSIFICATION TIME

Ontology	T_{HermiT}	T_{AR}	T_{MR}	T_{ComR}
Protein	53995	50	1229	1983(↓ 96.3%)
NCI	529442	19835	4990	26656(↓95.1%)
ICNP	9132	37	869	1053(↓ 88.5%)
OBI	6816171	3393157	941	3394672(↓ 50.2%)

EL module is based on the $\top\bot^*$-AD, which is a refinement w.r.t. set inclusion of both $\top$-AD and$\bot$-AD [22]. In our ComR system, ELK is chosen as main reasoner (MR) and the state-of-the-art HermiT acts as a the assistant reasoner (AR).

We have run the Algorithm 2 for task decomposition and Algorithm 3 for classification over several large bio-medical ontologies available from BioPortal [11]. The experimental results are summarised in Table I and Table II, respectively. All experiments were conducted on a laptop with an AMD quad core A6-3400 APU with 4GB of memory running Windows 7.

In Table I, the second column shows the size of the test ontology O, and the third column provides the number of non-EL axioms in the ontology, the fourth column presents the number of atoms in AD(O), and the final column gives the number of axioms in $O_{\overline{EL}}$ and the corresponding percentage.

Table II shows the classification time taken by HermiT and ComR, and the unit is millisecond. The second column shows time that the hermiT takes to classify the original ontology. The third and fourth columns present time taken by AR and MR, respectively. The final column gives the total classification time taken by ComR, which including T_{AR}, T_{MR}, and time used to constructed the temporary ontology $O^{'}$.

B. DISCUSSION

From statistical data in Table I and Table II, it is easy to see that the smaller the $O_{\overline{EL}}$ is, the more the reduced time is while classifying an ontology. The ontology Protein contains only 16 non-EL axioms and the size of the corresponding $O_{\overline{EL}}$ is very small, hence the bulk of workload is assigned to the main reasoner, and ComR significantly outperforms HermiT. OBI, However, contains a large number of axioms outside OWL 2 EL, thus the size of $O_{\overline{EL}}$ is proportionally much bigger and a higher workload is assigned to assistant reasoner, which results in a more modest performance gain.

From the literature, the closest approach is MORe [24], which proposed modular classification technique based directly on modules extraction. they showed the subsummers of the concepts in EL-signatures can be completely determined using the EL-reasoner, the subsummers of remaining concepts are computed using OWL 2 reasoner. In order to assign the bulk of the workload to the EL reasoner, They used a heuristics to compute a larger EL-signatures Σ^L and extract the EL module M_{Σ^L} w.r.t Σ^L. However, their method can not guarantee to get a maximal EL-signatures, hence the non-EL module $M_{\overline{\Sigma^L}}$ is not necessarily minimal. Compared with the experimental results in [24], our method (Algorithm 2) can identify a smaller non-EL module from the same tested ontologies, which are shown in Table III.

Table III
OUR WORK VS [24]. COMPARISON OF NON-EL MODULES

Ontology	$O_{\overline{EL}}$ in [24]	$O_{\overline{EL}}$ in our work
Protein	6.6%	0.62%
NCI	15.4%	3.3%

The reason is that extracting a module for a general signature is more complicated—axioms can pull in a module terms that are not strictly necessary for them to be nonlocal [22]. For example, only axiom α_4 in Example 1 is non-$\bot$-local w.r.t. Σ=$\{$Vergan$\}$. However, each module containing α_4 contains also α_1, α_2, and α_3, because in order to preserve the meaning of Vegan we need first to preserve the meaning of the other terms occurring in this axioms. To guarantee this condition, we need to enlarge Σ with the terms pulled in by relevancy, and then recheck the axioms against relevancy w.r.t. the new signature.

From the point of view of optimization, our method leads to a more reasonable task assignment—the bulk of workload is assigned to the efficient EL reasoner, and it can offer a bigger speedup in ontology classification, that is, our approach can significantly reduce time for ontology classification, which are shown in Table IV.

Table IV
OUR WORK VS [24]. COMPARISON OF REDUCED CLASSIFICATION TIME

Ontology	[24]	our work
Protein	$\downarrow$ 74.6%	$\downarrow$ 96.3%
NCI	$\downarrow$ 66.0%	$\downarrow$ 95.1%

VI. CONCLUSION AND OUTLOOK

In this paper, we have proposed a technique to combine OWL 2 reasoner with EL reasoner for SROIQ ontology classification. The combination is based on a suitable task assignment where the bulk of workload is assigned to the efficient EL reasoner, and the minimal non-EL module is given to OWL 2 reasoner. The technique makes users use expressive DLs to build their ontologies and still enjoy the efficient services as in tractable languages. Our method makes full use of the advantages of OWL 2 reasoner and EL reasoner. And the implementation is very simple. Although some time is taken to decompose the ontology before reasoning, the job of ontology decomposition can be handled off-line, and it is possible to maintain ontologies in a decomposed form [22].

We intend to continue our work in two directions. First, we will investigate the possibility of concurrent reasoning for the decomposed form of an ontology. Second, we will explore the ontology rewriting technique for the $O_{\overline{EL}}$, so that we can only use the efficient EL reasoners to classify an ontology expressed in expressive language.

ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China (No. 61070202) and National High Technology Research and Development Program of China (863 Program) (No. 2013AA013204).

REFERENCES

[1] B. Motik, P. F. Patel-Schneider, and B. Cuenca Grau, "OWL 2 Web Ontology Language Direct Semantics," *W3C Recommendation*, 2009.

[2] A. Sidhu, T. Dillon, E. Chang, and B. S. Sidhu, "Protein ontology development using OWL," in *Proc. of the OWL: Experiences and DirectionsWorkshop (OWLED 2005)*, volume 188 of CEUR Workshop Proceedings, Galway, Ireland.

[3] C. Golbreich, S. Zhang, and O. Bodenreider, "The foundational model of anatomy in OWL: Experience and perspectives," *Journal of Web Semantics*, 4(3), pp. 181-195, 2006.

[4] A. Rector and J. Rogers, "Ontological and practical issues in using a description logic to represent medical concept systems: Experience from GALEN," in *Reasoning Web, sencon international summer school, Springer*, 2006, pp. 197-231.

[5] Y. Kazakov, "RIQ and SROIQ are harder than SHOIQ," in *Proc. of Knowledge Representation and Reasoning (KR'08)*, 2008, pp. 274-284.

[6] B. Glimm, I. Horrocks, B. Motik, R. Shearer, and G. Stoilos, " A novel approach to ontology classification," *Journal of Web Semantics*, 14(1), pp. 84-101, 2011.

[7] V. Haarslev and R. Möller,"Racer system description," in *Proc. of IJCAR (IJCAR'01)*, 2001, vol. 2083, pp. 701-705.

[8] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL DL reasoner," *Journal of Web Semantics*, 5(2), 51-53, 2007.

[9] D. Tsarkov and I. Horrocks, "FaCT++ description logic reasoner: System description," in *Proc. of IJCAR (IJCAR'06)*, 2006, vol. 4130, pp. 292-297.

[10] Y. Kazakov and Markus Krözsch, "The Incredible ELK," http://elk.semanticweb.org.

[11] http://bioportal.bioontology.org

[12] F. Baader, S. Brandt, and C. Lutz, "Pushing the EL envelope," in *Proc. of IJCAI (IJCAI'05)*, 2005, pp. 364-369.

[13] J. Mendez and B. Suntisrivaraporn, "Reintroducing CEL as an OWL 2 EL Reasoner," in *Proc. of the International Workshop on Description Logics (DL'09)*, 2009, vol. 477.

[14] Mendez and Julian, "jCel: A modular rule-based reasoner," in *Proc. of the 1st Int. Workshop on OWL Reasoner Evaluation (ORE'12)*, 2012, vol. 858.

[15] Y. Kazakov, M. Krötzsch, and F. Simancík. "Concurrent classification of EL ontologies," in *Proc. of International Semantic Web Conference (ISWC'11)*, 2011, pp. 305-320.

[16] B.Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, et al. "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration," *Nature Biotechnology*, 25(11), 1251-1255, 2007.

[17] C. Del Vescovo, B. Parsia, U.Sattler, and T. Schneider, "The modular structure of an ontology: Atomic decomposition," in *Pro. of IJCAI (IJCAI'11)*, 2011, pp. 2232-2237.

[18] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler, "Owl 2: The next step for OWL," *Journal of Web Semantics*, 6(4), 309-322, 2008.

[19] I. Horrocks, O. Kutz and U. Sattler, "The even more irresistible SROIQ," in *Proc. of Knowledge Representation and Reasoning (KR'06)*, 2006, pp. 57-67,.

[20] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Modular reuse of ontologies: Theory and practice," *Journal of Artificial Intelligence Researchvol*, 31(1), pp. 273-318, 2008.

[21] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler, "Just the right amount: extracting modules from ontologies," in *Pro. of International Conference on World Wide Web (WWW'07)*, 2007, pp. 717-726.

[22] C. Del Vescovo, D. D. Gessler, P. Klinov, B. Parsia, U. Sattler, T. Schneider, and A. Winget, "Decomposition and modular structure of bioportal ontologies," in *Pro. of International Semantic Web Conference (ISWC'11)*, 2011, pp. 130-145.

[23] http : //owlapi.sourceforge.net/

[24] A. Armas Romero, B. Cuenca Grau, and I. Horrocks, "MORe: modular combination of OWL reasoners for ontology classification," in *Pro. of International Semantic Web Conference(ISWC'12)*, 2012, pp. 1-16.