

OPTIMIZING DESCRIPTION LOGIC REASONING FOR THE SERVICE MATCHMAKING AND COMPOSITION

Marian BABÍK, Ladislav HLUCHÝ

*Intelligent and Knowledge-based Technologies Group
Department of Parallel and Distributed Computing
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: Marian.Babik@saske.sk*

Manuscript received 20 August 2007; revised 14 February 2008
Communicated by Jacek Kitowski

Abstract. The Semantic Web is a recent initiative to expose semantically rich information associated with Web resources to build more intelligent Web-based systems. Recently, several projects have embraced this vision and there are several successful applications that combine the strengths of the Web and of semantic technologies. However, Semantic Web still lacks a technology, which would provide the needed scalability and integration with existing infrastructure. In this paper we present our ongoing work on a Semantic Web repository, which is capable of addressing complex schemas and answer queries over ontologies with large number of instances. We present the details of our approach and describe the underlying architecture of the system. We conclude with a performance evaluation, which compares the current state-of-the-art reasoners with our system.

Keywords: Semantic web, composition, matchmaking, description logic, reasoning

1 INTRODUCTION

Semantic Web is a recent effort, which tries to extend the current Web technologies by providing a well defined meaning to the services and information, thus enabling

computers and people to work in cooperation [24]. The Semantic Web is seen as a possible infrastructure, which can provide an environment for hosting and managing heterogeneous services. The use of the Semantic Web technologies in data integration and semantic web services is not novel. It has already been useful in providing a scalable solution for the planning and resource optimization of the Web service-based infrastructure [27]. However, as the number of resources and data increases, the creation and management of the heterogeneous and dynamic resources is the key to the future Web-based service-oriented computing.

The need for semantically enabled technologies was recognized in several scientific applications such as bioinformatics, chemistry and environmental sciences [34, 32, 20]. These applications require support for the dynamic and complex workflows, which are based on processing and sharing of large amounts of heterogeneous data. Recently, numerous projects have developed such workflows based on the composition and interoperability between grid and web services [20, 34, 23]. Such environments often require support for discovery, matchmaking, composition and executions of the grid and web services. One of the major obstacles for such systems is the lack of scalability of the existing semantic repositories, which are based on the Semantic Web standards such as RDF/RDFS or OWL.

In this paper we present a novel semantic repository, which can support the subset of the OWL standard and can thus address the service *matchmaking* and *composition* based on the semantic description of services.

2 MOTIVATION: SERVICE MATCHMAKING AND COMPOSITION

The Semantic Web is making available technologies, which support and to some extent automate knowledge sharing. In particular, there are several existing initiatives (OWL-S, WSMO, SWSL, SAWSDL¹), which provide evidence that ontologies with their ability to interweave human understanding of symbols with their machine-processability can play a key role in automating service matchmaking and composition in the Semantic Web. This is also supported by numerous successful extensions of the matchmaking and composition algorithms to the semantic web services [30, 6]. Generally, all semantic-based approaches share the idea that:

- At *publishing* time a set of relevant domain ontologies can be used to semantically annotate Web and Grid service descriptions (i.e. describing the capabilities of the services).
- At *matchmaking and composition* time, the same set of ontologies can be used to describe the functional criteria of the service, that the requester wishes to interact with or alternatively compose into a set of complex services. Hence,

¹ OWL-S: <http://www.daml.org/services/owl-s/1.1>, WSMO: <http://www.wsmo.org>, SWSL: <http://www.daml.org/services/swsl>, SAWSDL: www.w3.org/2002/ws/sawsdl

accessing the knowledge modeled in the ontologies, is not limited to syntactical matching, but it can also exploit the existing semantic matching.

Service *matching* is defined as the process when the service advertisement describes a service similar to the service request. The notion of similarity in the context of description logics is based on the subsumption of concepts, i.e. the key reasoning problem, which needs to be addressed in order to perform scalable *matchmaking* is the *terminological* reasoning [30]. This encompasses the decision whether the given set of concepts is in equivalent, disjoint or subsumption relation. In order to do that, it is necessary to compute a so called *classification* of the knowledge base (KB), i.e. to pair-wise compare the relationships of all the concepts and roles in KB . Determining the relationship between the concepts is then a matter of accessing such *classification* structures. Classification also allows to combine different ontologies by deciding on the actual relations between the concepts and roles of the KB .

Description logic languages describe services in a way amenable to planning (*composition*), since service has preconditions and effects that can be expressed as logical expressions. Using such similarity it is possible to use Web services as planning operators and to use a casual planner (e.g. HTN), to generate Web service compositions. The key reasoning problem for *composition* is then instance retrieval, which is used in two different ways. First, the evaluation of preconditions in operator and method descriptions, where the applicability of an action is determined by checking the precondition expression against the current state of the world. A second use is in task *matching*, which is similar to the service *matching* procedure described previously. Unlike service *matching*, task *matching* relies on determining query subsumption between precondition and effect expressions. During planning, the planner often evaluates many preconditions and tests for task *matching*, which can range over a large number of tasks (service methods). For these reasons the performance of the planning system is affected considerably by the query answering performance of the reasoner in both terminological and assertional reasoning problems.

In summary, our goal is to provide a scalable ontological repository for services and data, which would support both terminological and assertional queries as well as coupling with legacy metadata stored in relational databases.

3 APPROACH

This section outlines our approach to reasoning in the description logic *SHIQ*, which is very interesting for our purposes as it represents a significant subset of the OWL-DL standard. OWL-DL is based on the class of description logic *SHOIN(D)*; the main difference to *SHIQ* is the possibility to express nominals [16]. Our approach is based on two fundamental ideas: firstly, we rely on the well-known semantics of the existing decision procedures; secondly, we combine the existing optimizations of the procedures to create a more scalable and effective system. Our method is based on the well known tableau-based decision procedure [18] and reasoning in the framework

of resolution [19]. The tableau calculus provides very effective handling of complex description logic schemas (TBoxes). The resolution-based approach is based on the translation of the description logic to the disjunctive datalog and provides extensive coverage for handling reasoning with large number of individuals (ABox).

To support a storage and inference system for large scale OWL ontologies on top of relational databases we have developed a new approach with the following characteristics:

- Our method combines the existing description logics reasoners for computing taxonomies (TBoxes), i.e. TB , with rule-based reasoners for reasoning with large number of instances (ABoxes), i.e. DD .
- Based on the proposed combination we can re-use the existing optimizations (i.e. classification and satisfiability techniques) of the description logics reasoners to perform fast classifications of the complex schemas. Further, we can exploit the optimizations of the rule-based systems (i.e., join-order and magic sets) to perform queries over ontologies with large number of instances. Since deductive databases are designed to perform the queries over existing relational databases, it is possible to integrate our system with existing RDBMS-based registries.

The key theoretical aspect of our approach lies in the definition of the syntactical variant of the description logics and in description of how the respective description logics reasoning problems are solved within our system.

3.1 Expressive Description Logics

Following our previous overview we will define the subset of the expressive description logics for which our method is suitable, i.e. a $SHIQ$ class of description logics.

Definition 1 ($SHIQ$ RBox). Let N_R be the set of abstract role names; then the set of abstract roles is $N_R \cup R^- | R \in N_R$, where $Inv(R) = R^-$ and $Inv(R^-) = R$. A $SHIQ$ RBox KB_R is a finite set of transitivity axioms $Trans(R)$ and abstract role inclusion axioms $R \sqsubseteq S \in KB_RBox$.

Further, R is a simple role if there is no role S such that $S \sqsubseteq R$ and S is transitive. R is a complex role if it is not simple. The set of possible $SHIQ$ concepts is defined as follows:

Definition 2 ($SHIQ$ concepts). Let N_C be the set of atomic concepts, then the set of $SHIQ$ concepts over N_R and N_C is defined inductively as the minimal set for which the following holds:

1. \top and \perp are $SHIQ$ concepts.
2. Each atomic concept $A \in N_C$ is a $SHIQ$ concept.
3. If C, D are $SHIQ$ concepts, R is an abstract role, S is an abstract simple role, then $\neg C, C \sqcup D, C \sqcap D, \exists R.C, \forall R.C, \leq S.C$ and $\geq S.C$ are also $SHIQ$ concepts.

Concepts that are not in N_C are called complex concepts.

Definition 3 (*SHIQ TBox*). A *SHIQ* TBox KB_T over N_C and KB_R is a finite set of concept inclusion axioms (GCI) $C \sqsubseteq D$ or concept equivalences $C \equiv D$, where C, D are *SHIQ* concepts.

Definition 4 (*SHIQ ABox*). Let N_I be a set of abstract individuals. A *SHIQ* ABox KB_A is a set of concept and role membership axioms of the form $C(a), R(a, b)$ and equality axioms $a = b, a! = b$, where C is a *SHIQ* concept, R is an abstract role and a, b are individuals.

Definition 5 (*SHIQ Knowledge Base*). A *SHIQ* knowledge base KB is a triple (KB_R, KB_T, KB_A) , where KB_R is a *SHIQ* RBox, KB_T is a *SHIQ* TBox and KB_A is a *SHIQ* ABox and the sets N_R, N_I, N_C are mutually disjoint.

The above definitions follow the typical definitions of description logics knowledge bases [1]. The definitions differ mainly in the aspect of restricting the complex roles in the negative membership axioms. This is due to the limitation of the resolution-based method for the *SHIQ* knowledge bases. The previous definitions can be extended in a straight-forward manner to cover concrete domains, i.e. *SHIQ(D)* [1]. Reasoning with concrete domains can also be supported as both resolution and tableau-based method support such extension.

The semantics of the *SHIQ* KB can be defined by the direct-model theoretic semantics or by the transformation of the axioms into a first-order formula [18, 4]. It was shown by [4] that both semantics coincide. Our approach is based on the proof of [19], which identifies the subset of description logics for which it can be shown that both tableau and resolution-based approaches derive the same set of consequences given the same *SHIQ* KB .

In the next section we provide a detailed overview of the description logics reasoning problems and show how it can fit into the context of combined decision procedures.

3.2 Reasoning Problems

In the previous sections we have defined the *SHIQ* KB and provided an overview of our approach to description logics reasoning. This section will provide a detailed overview of possible queries and inferences and show how the actual queries are realized in the running system. All of the described reasoning tasks are part of a larger set of possible DL reasoning queries, which we will denote Q_{DL} . These tasks can be divided into two specific areas, i.e. assertional (ABox) and terminological (TBox) reasoning. For the ABox reasoning we will rely on the translation of the *SHIQ* KB to the disjunctive datalog program, which we will denote $DD(KB)$. For the TBox reasoning we will rely on the satisfiability test *SAT* implemented as an optimized tableau calculus.

3.2.1 Assertional Reasoning

The expressivity of the actual queries is determined by the underlying knowledge base, i.e. KB . In our case we can perform all possible *SHIQ* queries without any restrictions.

Instance checking is a problem of determining whether an individual a is an individual of concept C . A similar reasoning problem for roles is called ground role fillers, i.e. determining whether two individuals a, b are related with a role R . The query is equivalent to the instance retrieval of concepts and thus can be performed in a similar manner. In our case depending on the nature of the concept C there can be two possibilities:

Corollary 1. Let KB be a *SHIQ* knowledge base; then:

1. $KB \models \alpha$ iff $DD(KB) \models \alpha$, where α is of the form $A(a), R(a, b)$ and A is an atomic role.
2. $KB \models C(a)$ for a non-negated concept C and an individual a , iff $DD(KB \cup \{C \sqsubseteq Q\}) \models Q(a)$, where Q is a new atomic concept.

Instance retrieval is a problem of determining all individuals of the given concept C . A non-optimized algorithm for a retrieval query can be realized by testing for each individual occurring in the KB , whether it is an instance of the concept C (performing n instance checking queries, where n is the number of individuals in the KB). Since $DD(KB)$ is based on the translation to disjunctive datalog, the query can be performed simply by checking whether $DD(KB) \models C(\theta(\mathbf{x}))$, where θ is an assignment of individuals to variables. The result is then obtained by accumulating the individuals of a given variable.

Dual retrieval is a problem of determining all concepts which a given individual a instantiates. This amounts to asking a number of instance checking queries, i.e. for n concepts in KB this results in n $DD(C(a))$ queries and accumulating the positive answers. This process can be optimized by taking into consideration the subsumption relationship between concepts, and thus it reuses the optimizations, which are described in Section 3.2.2.

Conjunctive queries are a special form of assertional queries, which were introduced as a formalism to express the class of select-from queries known from the relational databases. Algorithms for answering conjunctive queries are usually based on the reduction of the query answering to well known description logics reasoning problems, thus re-using the existing reasoning systems. Hence, conjunctive query answering can be seen as a special formalism representing the DL reasoning problems (mostly assertional queries) in the form of a relational query. In our approach we follow the query answering in the framework of resolution [19].

Definition 6. Let KB be a *SHIQ* knowledge base and let x_1, \dots, x_n and y_1, \dots, y_m be a set of distinguished and non-distinguished variables, denoted $\{\mathbf{x}, \mathbf{y}\}$. A conjunctive query over KB , denoted $Q(\{\mathbf{x}, \mathbf{y}\})$, is a finite set of conjunctions of DL axioms of the form $A(s)$ or $R(s, t)$, where A is an atomic concept, R is an abstract role and s, t are individuals from the KB or variables from $\{\mathbf{x}, \mathbf{y}\}$.

The reasoning problems of the conjunctive queries are realized as follows:

Corollary 2. An answer of a query $Q(\{\mathbf{x}, \mathbf{y}\})$ w.r.t KB is an assignment θ of individuals to distinguished variables such that $DD(KB) \models \exists y : Q(\{\mathbf{x}\theta\}, \{\mathbf{y}\})$.

3.2.2 Terminological Reasoning

One of the disadvantages of the resolution-based procedure is that optimizing query answering of the *TBox* related problems is a complex domain of the *DL* reasoners and there is no straightforward transformation of these optimization techniques to the $DD(KB)$. This results in overall better performance of the *DL* reasoners over *DD* reasoner in terms of the *TBox* queries. The tableau-based reasoners rely on the ability to compute the satisfiability procedure denoted *SAT*.

Concept satisfiability (*SAT*) is the process of determining whether the given concept is satisfiable, i.e. whether $KB \cup \{C\}$ is satisfiable. Following the tableau-based procedure, the concept satisfiability can be decided by checking whether $KB \cup \{C(q)\}$ is consistent, where q is an arbitrarily chosen individual name (i.e., random individual not related to *ABox*). The consistency of the formula can be decided with the tableau decision procedure [18].

Subsumption of concepts is a problem of determining whether one concept is subsumed by another, i.e. determining whether the axiom $C \sqsubseteq D$ holds. In our case we can perform the subsumption check following the definition [1]:

Corollary 3. Let C, D be *SHIQ* concepts; then $C \sqsubseteq D \Leftrightarrow C \sqcap \neg D$ is unsatisfiable.

Hence, the subsumption of C, D can be computed as $\neg SAT(KB \cup \{C(z) \sqcap \neg D(q)\})$, where z, q are arbitrarily chosen individual names.

The consequences of the previous section also hold in terms of concept equivalence. The equivalence of the concepts is defined as follows:

Corollary 4. Let C, D be *SHIQ* concepts, then $C \equiv D \Leftrightarrow C \sqcap \neg D$ and $\neg C \sqcap D$ are unsatisfiable.

The equivalence can thus be reduced to checking if $\neg SAT(KB \cup \{C(z) \sqcap \neg D(q)\})$ and $\neg SAT(KB \cup \{\neg C(z) \sqcap D(q)\})$, where z, q are arbitrarily chosen individual names.

Disjointness of concepts follows the definition of the subsumption and equivalence, i.e.:

Corollary 5. Let C, D be *SHIQ* concepts; then C, D are disjoint $\Leftrightarrow C \sqcap D$ is unsatisfiable.

The disjointness of concepts can thus be reduced to checking if $\neg SAT(KB \cup \{C(z) \sqcap D(q)\})$, where z, q are arbitrarily chosen individual names.

While the previous definitions have been focused on the *SHIQ* concepts, the same approach can be applied to the *SHIQ* abstract roles.

Complete classification of the hierarchy of complex concepts and roles means to pairwise compare the relationships of all the concepts and roles. Since the resolution-based procedures are not optimized for satisfiability checking and thus result in systems which have to rely on the previously defined approaches, computing complete classification would result in performing $n(n-1)$ subsumption checks for n concepts. The overall complexity of classification is thus $O(n^2)$. Although there are various possible optimizations from the DL reasoners, which can be reused in the *LP* systems, the overall performance is still not satisfactory. Most of the optimizations are based on the traversal of the *KB* and reduction of the number of tests that are required to compute the classification. The most prominent optimizations are top, bottom search and told subsumption [1].

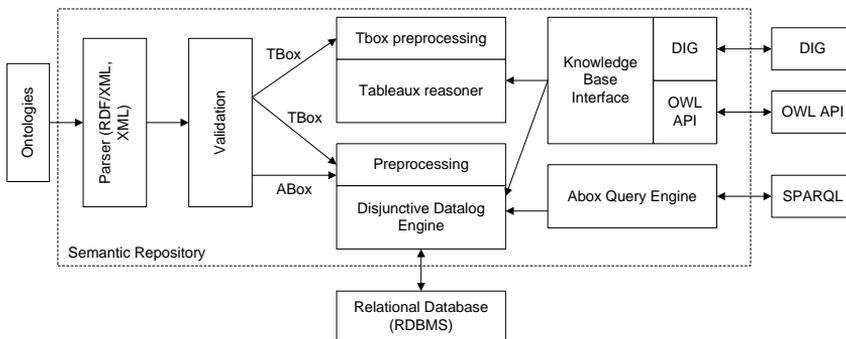


Fig. 1. An overview of the architecture

4 IMPLEMENTATION

Overall architecture of the system is based on extending the existing *SHIQ* tableau reasoning strategy with the optimizations for the conjunctive query answering and relational database backend. Figure 1 shows the main components of the system. The core of the system is composed of two reasoners, tableau reasoner and disjunctive

datalog engine. The aim of the tableau reasoner is to check the consistency of the *TBox* and to compute its classification. Disjunctive datalog engine is based on the KAON2 [19] and its aim is to check the consistency of the knowledge base KB^2 and to perform the conjunctive queries over *ABox*.

The architecture of the system is shown in Figure 1. The primary role of the system is to perform ontology reasoning, storage and retrieval of the ontological axioms, and the integration with relational databases (RDBMS). The architecture is composed of the four main components, the parsing and preprocessing of the ontologies is performed by the RDF/XML parser and validation component. The inferencing capabilities are provided by the tableau reasoner and disjunctive datalog engine, which can query the instances stored in the relational database (RDBMS). The knowledge base interface provides a common interface for storage, retrieval functions and inferencing capabilities of the system. On top of the knowledge base interface there are OWL and JENA APIs [3, 7]. The ABox Query Engine provides support for the SPARQL queries [10]. The external services can access the system by calling the methods of the respective interfaces, i.e. OWL API, SPARQL or Jena API.

The OWL ontologies are loaded into the reasoners after parsing and validation. The validation ensures that all resources are valid and the actual expressivity is within the boundaries of our method. During the loading phase, axioms about classes and properties are put into *TBox* and assertions about instances are stored in the *ABox*. *TBox* axioms are then preprocessed and fed into the tableau reasoner. Additionally, *TBox* is preprocessed for the resolution method and, together with the *ABox*, loaded into the disjunctive datalog engine. The engine performs the necessary preprocessing and clausification of the KB . The outcome of such process is a disjunctive datalog program, which can be used to answer conjunctive queries. Although the transformation to $DD(KB)$ is quite complex, it is performed only once during the initialization of the KB . The set of rules of the $DD(KB)$ can be saved for later re-use. Alternatively, *ABox* assertions can be stored in the relational database (RDBMS): by mapping ontology entities to database tables, disjunctive datalog engine can then be used to query the database on the fly during the reasoning.

5 PERFORMANCE EVALUATION

In this section we present a comparison of the performance of query answering of our approach with tableau and resolution based reasoners. This should provide an insight into practical applicability of our approach.

It should be noted that due to a large number of optimizations and high complexity of the methods, it is very difficult to separate the reasoning methods from their respective implementations. There are numerous low-level optimizations that are implemented in the methods and the choice of e.g. data structures or memory

² The consistency of the knowledge base can only be decided based on the overall *TBox* and *ABox*.

management can easily dominate the reasoning time. Furthermore, the implementations are written in different languages and usually provide only proprietary source code, which makes the evaluation even more complex. Therefore the results of the section present an overview of the performance, which can be expected in the real-life scenarios, rather than a definitive measure of the complexity of the reasoning algorithms.

5.1 Test Settings

We have compared the performance of our approach (denoted Hires) with the three recent description logics reasoners, KAON2, Pellet and Racer [19, 13, 31]. We did not consider the other description logics reasoners due to their limitations; Fact, Fact++ and DLP do not support ABox reasoning [15, 28], LOOM is incomplete [26], CLASSIC, OWLIM, Minerva, JENA support only a subset of the SHIN(D) description logics [7, 5, 35, 21].

The sequence of API calls that we used for each reasoner were determined from the LUBM benchmarks. We have tried to accommodate all the recommendations for the performance evaluation of each reasoner. For each reasoning task we have started a fresh instance of the reasoner and loaded the test knowledge base. This was done mainly due to significant problems with memory management of the reasoners during repetitive querying [33]. We have measured the time required to execute the task. We have assured that all systems returned the same answers.

In case of the tableau-based reasoners the optimizations of the ABox queries usually involve caching of computation results, thus performance can increase with subsequent queries. Further, both Racer and Pellet check the ABox consistency during which they compute the index for the instance retrieval, which affects its initial performance severely. Since we have not considered any caching and materialization techniques in our approach, we have measured both the time for ABox consistency and the time to answer the query. It should be noted that computing the index for the instance retrieval is not feasible in many applications due to a large number of individuals. In case of KAON2 we have not measured the time to compute the datalog program as it was insignificant.

All tests were performed on a laptop computer (T60) with 1.8 GHz memory and 1 GB of RAM, running Linux kernel 2.6.20-1. For Java-based reasoners (Pellet, KAON2) we have used Java runtime 1.5.0 Update 6 with virtual memory restricted to 800 MB. We run each reasoning task five times and plotted the average of the set. Each task had a time limit of 5 minutes. Tests that run out of memory are denoted with a dash (-). Tests that run out of time are denoted with time 300 000.

5.2 Test Ontologies

We have based our test on the existing benchmark ontologies for the ABox and TBox reasoning as well as on the realistic ontologies developed within the Semantic Web Community. In order to obtain sufficient number of individuals we have performed

ABox replication, i.e. duplication and renaming of the individuals in the ABox. Table 1 shows the statistics about the structure and complexity of the ontologies.

KB	$C \sqsubseteq D$	$C \equiv D$	$C \sqcap D \sqsubseteq \perp$	RBox	$C(a)$	$R(a, b)$
semintec_1	55	0	113	6	17 941	47 248
semintec_2					35 882	94 496
semintec_3					53 823	141 744
semintec_4					71 764	188 992
semintec_5					89 705	236 240
vicodi_1	193	0	0	10	16 942	36 711
vicodi_2					33 884	73 422
vicodi_3					50 826	110 133
vicodi_4					67 768	146 844
lubm_1	36	6	0	9	18 128	49 336
lubm_2					40 508	113 463
lubm_3					58 897	166 682
lubm_4					83 200	236 514
wine_8	126	61	1	9	20 007	19 926
wine_9					39 767	39 606
wine_10					79 287	78 966
owls_1	82	5	12	9	50 000	0
owls_2					75 000	0
owls_3					120 000	0
dolce	203	27	42	522	0	0
galen	3 237	699	0	287	0	0

Table 1. Statistics of the benchmark ontologies

5.3 Performance tests

VICODI. Since this ontology contains only very simple TBox, it can be expected that resolution-based decision procedure will dominate the test with increasing number of individuals. We have performed the following conjunctive queries over the Vicodi ontology:

$$Q_1(x) = \text{Individual}(x)$$

$$Q_2(x, y, z) = \text{Military} - \text{Person}(x), \text{hasRole}(x, y), \text{related}(x, z).$$

Further we have performed a classification of the TBox (Q_3). The results in Figure 2 show that Pellet and Racer are dominant in answering Q_3 , while KAON2 performs better on the Q_1, Q_2 queries. This is due to its connection with the deductive database, which plays a key role in answering conjunctive queries for a simple TBox. It can be seen that while both Pellet and Racer employ the same strategy for answering queries, there is a gap between their performances. This is mainly due to various different optimizations, which in the case of Racer

are dominant for the classification, and in the case Pellet are dominant for the conjunctive query answering.

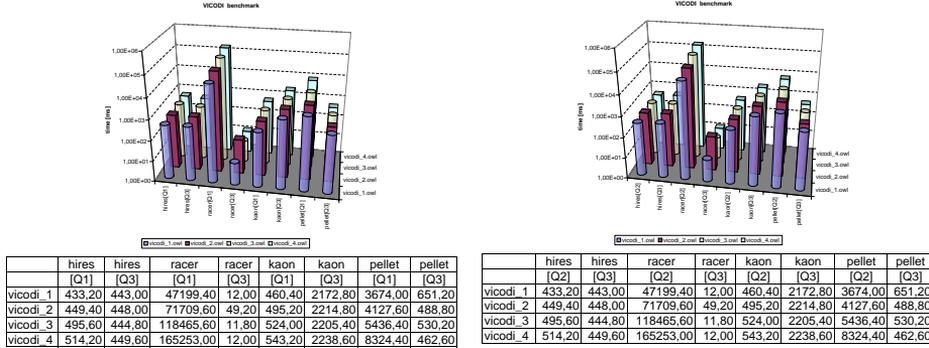


Fig. 2. Experimental results for the VICODI ontology, Q_1, Q_3 (left) and Q_2, Q_3 (right)

SEMINTEC. Similarly to the previous case, Semintec is also a very simple ontology; thus the results can be expected to follow the same pattern. Unlike Vicodi, Semintec contains functional roles, which are more difficult for the deductive databases. We have performed the following queries:

$$Q_1(x) = \text{Person}(x)$$

$$Q_2(x, y, z) = \text{Man}(x), \text{isCreditCardOf}(y, x), \text{Gold}(y), \\ \text{livesIn}(x, z), \text{Region}(z).$$

We have also performed classification as a query (Q_3). The results are shown in Figure 3. It can be seen that while the performance deteriorates for KAON2 and Hires, in conjunctive query answering both still outperform Pellet and Racer. In terms of the TBox classification the situation is contrary to the conjunctive queries, with Racer being the best performer followed by Pellet and Hires.

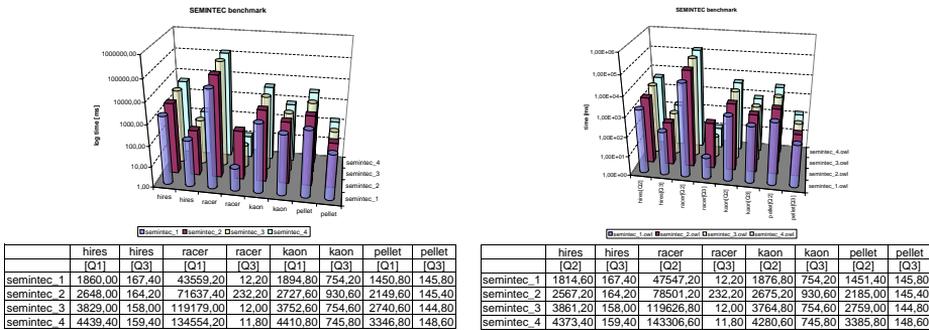


Fig. 3. Experimental results for the SEMINTEC ontology, Q_1, Q_3 (left) and Q_2, Q_3 (right)

Lehigh University Benchmark (LUBM) is comparable to Semintec and Vi-codi in terms of size; however, it contains more complex TBox concepts. Since the original benchmark contains several queries for which we have had similar results, we have chosen a set of simple and complex queries:

$$\begin{aligned}
 Q_1(x) &= \text{Chair}(x) \\
 Q_2(x, y) &= \text{Chair}(x), \text{worksFor}(y), \text{Department}(z), \\
 &\quad \text{subOrganizationOf}(y, \text{http://www.University0.edu}) \\
 Q_3(x, y, z) &= \text{Student}(x), \text{Faculty}(y), \text{Course}(z), \text{advisor}(x, y), \\
 &\quad \text{takesCourse}(x, z), \text{teacherOf}(y, z).
 \end{aligned}$$

The results are shown in Figures 4 and 5. The performance of Hires and KAON2 follows the previous case. The performances of Racer and Pellet differ as Pellet outperforms Racer in answering conjunctive queries.

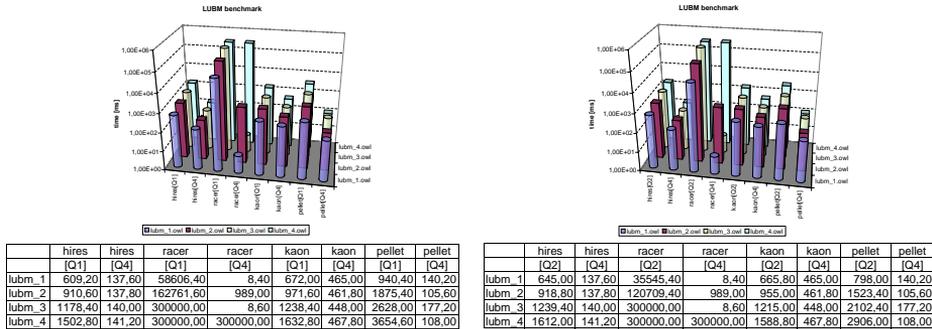


Fig. 4. Experimental results for the LUBM benchmark, Q_1, Q_4 (left) and Q_2, Q_4 (right)

OWL-S provides one of the most frequently used ontologies in the field of semantic web services. It contains a quite complicated TBox and can be connected to many other ontologies, which provide the domain model of the application. In our case we have performed the queries over a set of ontologies developed within the project K-WfGrid [20]. Apart from classification (Q_2) we have performed the following query:

$$Q_1(x) = \text{ServiceProfile}(x).$$

The results are shown in Figure 5. Extracting the existing *ServiceProfiles* is one of the most frequent queries in both composition and matchmaking of services as both rely on the ontological model of the *ServiceProfile*. The performance of reasoners reflects the increased complexity of the ontology.

Wine. Even more complex ontology than OWL-S is the Wine ontology, as it contains multiple disjunctions, which significantly affects the performance of both tableau and resolution based methods. We have performed the following query:

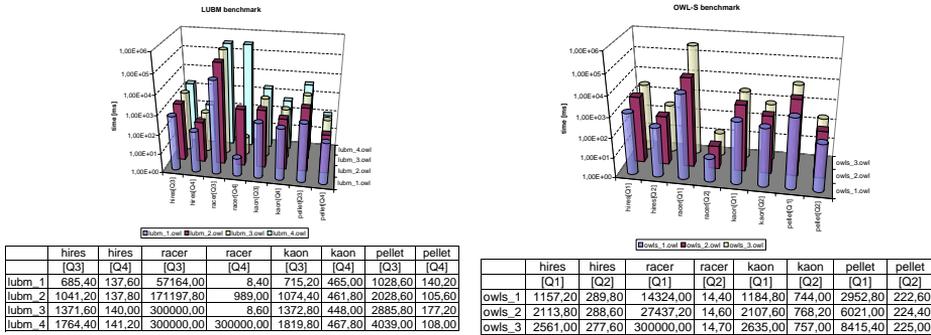


Fig. 5. Experimental results for the LUBM (Q_3, Q_4) (left) and OWL-S (right) ontologies

$$Q_1(x) = AmericanWine(x).$$

The results are shown in Figure 6. Increased complexity of the TBox affects the performance of all the reasoners severely. Furthermore, there is an increasing gap between classification of the TBox between tableau and resolution based methods.

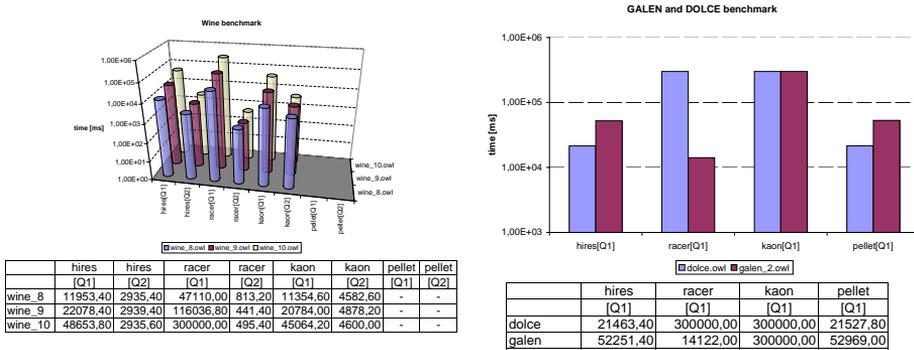


Fig. 6. Experimental results for the Wine (left) and Dolce, Galen (right) ontologies

GALEN and DOLCE are providing the most complex TBox, which can be handled by the existing reasoners. Unlike previous test, DOLCE and GALEN were used only in the context of the classification benchmark. The results are shown in Figure 6. Due to the extensive use of the transitive properties, KAON2 is unable to compute the classification in the given time frame. It is clear that the performance of KAON2 lags behind the tableau based methods. Similar results are also shown for the GALEN ontology.

We have performed extensive test of the performance on the ontologies from the Semantic Web community. Hires was shown to outperform all of the existing rea-

soners in terms of the combined TBox and ABox reasoning. While the performance of the tableau based methods was dominated in the TBox reasoning problems, the resolution based method dominates the conjunctive query answering and ABox reasoning problems. We have discussed the primary causes for this as a consequence of the particular optimizations.

It is clear that investigation of the possible hybrid approaches to the resolution and tableau based methods can bring interesting results. While our method was focused towards the use in the domain of semantic web services, it can be used to perform reasoning for the other fields of the Semantic Web research as well. This is mostly due to the distinctive combination of the methods with strict semantic integration.

6 RELATED WORK

This work was largely motivated by our previous experience in reasoning for the Semantic Web and Grid services [22, 23], which provided support for the grid-level ontological management, semantic metadata and semantic web services.

The current research in the description logics reasoning can be divided into two categories, i.e. tableau-based decision procedures and integration/reduction of description logics to other formalism. The design of the decision procedures for *SHOIN* and *SHIQ* has been accomplished only very recently [18]. Expressive description logics are known to have very high worst-case complexity. This suggests that there is a significant gap between design of the decision procedures and practical implementation achievements. Hence, naive implementations of the procedures are often useless in practice. Consequently, there are a large number of optimizations of the tableau-based decision procedure in order to achieve acceptable performance [17, 14, 11]. These optimizations lead to a significant improvement, mainly in the scope of terminological reasoning. Our approach extends the tableau-based decision procedure with instance retrieval optimizations.

In the category of integration or reduction to other formalisms, the most recent approaches are trying to integrate description logics with a rule-based system, e.g. answer set programming, deductive databases, etc. Such integration can be based on strict semantic separation, such as [9, 25]. These works mainly aim at extending the description logics with non-monotonic reasoning, fuzzy or probabilistic extensions, as well as arbitrary mixing of closed and open world reasoning. Another choice is to provide frameworks with strict semantic integration. [12] shows how to reduce reasoning with a subset of *SHIF* to an inference in Horn programs and vice versa. Other works such as [8, 29] represent hybrid approaches using description logics. These works present various combinations of datalog and descriptions logics *ALC* and *ALCNR*. The recent work of [19] presents a reduction of *SHIQ* ontologies to positive disjunctive datalog and proposes a novel resolution-based reasoning algorithm for the description logics. Unlike the above-mentioned work, our approach benefits from its integration with

tableau-based calculus and thus can provide more efficient terminological reasoning.

7 CONCLUSION

We have described design and development of the scalable semantic web repository. Currently, we are working on the evaluation of the proposed system on a real-life application, following the use case that we have developed during the project Knowledge Workflow Grid [2]. In the future we would like to address the caching and materialization aspects of the semantic storage.

Acknowledgments

The research reported in this paper has been partially financed by the Slovak national projects, NAZOU SPVV 1025/2004, SEMCO-WS APVV-0391-06, APVV RPEU-0024-06, VEGA 2/6103/6, VEGA 2/7098/27.

REFERENCES

- [1] BAADER, F.—CALVANESE, D.—MCGUINNESS, D. L.—NARDI, D.—PATEL-SCHNEIDER, P. F. eds.: *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [2] BABÍK, M.—HABALA, O.—HLUCHÝ, L.—LACLAVÍK, M.: *Semantic Services Grid in Flood-Forecasting Simulations*. *Computing and Informatics*, Vol. 26, 2007, No. 4, pp. 447–464.
- [3] BECHHOFFER, S.—VOLZ, R.—LORD, P.: *Cooking the Semantic Web with the OWL API*. In *ISWC 2003*, Sanibel Island, Florida, USA, October 2003.
- [4] BORGIDA, A.: *On the Relationship Between Description Logic and Predicate Logic Queries*. In *CIKM '94: Proceedings of the Third International Conference on Information and Knowledge Management (New York, NY, USA, 1994)*, ACM Press, pp. 219–225.
- [5] BORGIDA, A.—BRACHMAN, R. J.—MCGUINNESS, D. L.—RESNICK, L. A.: *CLASSIC: A Structural Data Model for Objects*. In *SIGMOD '89: Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data (New York, NY, USA, 1989)*, ACM, pp. 58–67.
- [6] CARDOSO, J.—SHETH, A.: *Semantic Web Services and Web Process Composition*. *First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004, Revised Selected Papers*, Vol. 3387, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [7] CARROLL, J. J.—DICKINSON, I.—DOLLIN, C.—REYNOLDS, D.—SEABORNE, A.—WILKINSON, K.: *Jena: Implementing the Semantic Web Recommendations*. In *WWW Alt. '04, Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters (2004)*, ACM Press, pp. 74–83.

- [8] DONINI, F. M.—LENZERINI, M.—NARDI, D.—SCHAERF, A.: AL-log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems*, Vol. 10, 1998, No. 3, pp. 227–252.
- [9] EITER, T.—LUKASIEWICZ, T.—SCHINDLAUER, R.—TOMPITS, H.: Combining Answer Set Programming with Description Logics for the Semantic Web. In *Proc. of the International Conference of Knowledge Representation and Reasoning (KR04)*, 2004.
- [10] FEIGENBAUM, L.—TORRES, E.—CLARK, K.: SPARQL Protocol for RDF. W3C recommendation, W3C, January 2008. <http://www.w3.org/TR/2008/REC-rdf-sparql-protocol-20080115/>.
- [11] GOCZYLA, K.—GRABOWSKA, T.—WALOSZEK, W.—ZAWADZKI, M.: The Knowledge Cartography – A New Approach to Reasoning over Description Logics Ontologies. In *SOFSEM 2006: Theory and Practice of Computer Science, 32nd Optimizing Description Logic Reasoning Conference on Current Trends in Theory and Practice of Computer Science*, Merín, Czech Republic, January 21–27, 2006, *Proceedings (2006)*, J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, J. Stuller (Eds.), Vol. 3831 of *Lecture Notes in Computer Science*, Springer, pp. 293–302.
- [12] GROSSOF, B.—HORROCKS, I.—VOLZ, R.—DECKER, S.: Description Logic Programs: Combining Logic Programs with Description Logic. In *Proceedings of WWW 2003*, Budapest, Hungary, May 2003.
- [13] HAARSLEV, V.—MÖLLER, R.: Description of the Racer System and Its Applications. In *Proceedings International Workshop on Description Logics (DL-2001)*, Stanford, USA, August 1-3, 2001, pp. 131–141.
- [14] HAARSLEV, V.—MÖLLER, R.: Optimization Strategies for Instance Retrieval. In *Proc. of the International Workshop on Description Logics*, Toulouse, France, 8390, 2002.
- [15] HORROCKS, I.: The FaCT System. In *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, Vol. 1397, 1998, pp. 307–350.
- [16] HORROCKS, I.—PATEL-SCHNEIDER, P.: Reducing OWL Entailment to Description Logic Satisfiability. In *Proc. of the 2nd International Semantic Web Conference (ISWC)*, 2003.
- [17] HORROCKS, I.—SATTTLER, U.: Optimised Reasoning for SHIQ. *Tech. Rep. LTCS-01-08*, Germany, 2001.
- [18] HORROCKS, I.—SATTTLER, U.: A Tableau Decision Procedure for SHOIQ. *Journal of Automated Reasoning*, Vol. 39, 2007, No. 3, pp. 249–276.
- [19] HUSTADT, U.—MOTIK, B.—SATTTLER, U.: Reasoning for Description Logics around SHIQ in a Resolution Framework. *Tech. Rep. 3-8-04/04*, FZI, Germany, 2004.
- [20] K-Wf Grid. K-Wf Grid Technical Annex. <http://www.kwfggrid.net>, 2004.
- [21] KIRYAKOV, A.—OGNYANOV, D.—MANOV, D.: OWLIM – A Pragmatic Semantic Repository for OWL. In *Web Information Systems Engineering WISE 2005 Workshops*. IEEE Computer Society, 2005, pp. 182–192.
- [22] KRYZA, B.—SŁOTA, R.—MAJEWSKA, M.—PIECZYKOLAN, J.—KITOWSKI, J.: Grid Organizational Memory-Provision of a High-Level Grid Abstraction Layer Sup-

- ported by Ontology Alignment. *Future Generation of Computer Systems*, Vol. 23, 2007, No. 3, pp. 348–358.
- [23] LACLAVÍK, M.—BALOGH, Z.—BABÍK, M.—HLUCHÝ L.: AgentOWL: Semantic Knowledge Model and Agent Architecture. *Computing and Informatics*, Vol. 25, 2006, No. 5, pp. 441–439.
- [24] LEE, B. T.—HENDLER, J.—LASSILA, O.: The Semantic Web. *Scientific American*, May 2001.
- [25] LUKASIEWICZ, T.: Probabilistic Description Logic Programs. *International Journal on Approximate Reasoning*, Vol. 45, 2007, No. 2, pp. 288–307.
- [26] MACGREGOR, R. M.: Inside the LOOM Description Classifier. *SIGART Bull.*, Vol. 2, 1991, No. 3, pp. 88–92.
- [27] MOTTA, E.—DOMINGUE, J.—CABRAL, L.—GASPARI, M.: IRS-II: A Framework and Infrastructure for Semantic Web Services. In *The Semantic Web — ISWC2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20–23, 2003, Proceedings (2003)*, D. Fensel, K.P. Sycara, J. Mylopoulos (Eds.), Vol. 2870 of *Lecture Notes in Computer Science*, Springer.
- [28] PATEL-SCHNEIDER, P.: DLP System Description. *Automated Deduction – CADE-17 1831/2000 (2000)*, pp. 297–301.
- [29] ROSATI, R.: DL+log: Tight Integration of Description Logics and Disjunctive Datalog. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning, 2006*, pp. 68–78.
- [30] SHVAIKO, P.—EUZENAT, J.: A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics, IV 3730 (2005)*, pp. 146–171.
- [31] SIRIN, E.—PARSIA, B.—GRAU, B. C.—KALYANPUR, A.—KATZ, Y.: Pellet: A Practical OWL-DL Reasoner. *Journal of Web Semantics*, Vol. 5, 2007, No. 2, pp. 51–53.
- [32] SUDHOLT, W.—ALTINTAS, I.—BALDRIDGE, K.: Scientific Workflow Infrastructure for Computational Chemistry on the Grid. In *International Conference on Computational Science, Vol. 3, 2006*, V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Vol. 3993 of *Lecture Notes in Computer Science*, Springer, pp. 69–76.
- [33] WEITHONER, T.—LIEBIG, T.—LUTHER, M.—BOHM, S.: Whats Wrong with OWL Benchmarks? In *Proc. of the Second Int. Workshop on Scalable Semantic Web Knowledge Base Systems, SSWS 2006, Athens, GA, USA, November 2006*, pp. 101–114.
- [34] WROE, C.—GOBLE, C.—GREENWOOD, M.—LORD, P.—MILES, S.—PAPAY, J.—PAYNE, T.—MOREAU, L.: Automating Experiments Using Semantic Data on a Bioinformatics Grid. *IEEE Intelligent Systems*, Vol. 19, 2004, No. 1, pp. 48–55.
- [35] ZHOU, J.—MA, L.—LIU, Q.—ZHANG, L.—YU, Y.—PAN, Y.: Minerva: A Scalable OWL Ontology Storage and Inference System. In *The Semantic Web, ASWC2006, First Asian Semantic Web Conference, Beijing, China, September 3–7, 2006, Proceedings (2006)*, *Lecture Notes in Computer Science*, Springer, pp. 429–443.



Marián BABÍK is a researcher at the Institute of Informatics of the Slovak Academy of Sciences at Center for Intelligent Technologies (CIT) and a member of the computing group at the Institute of Experimental Physics, SAS. In 2003 he received his M. Sc. degree in artificial intelligence from the Technical University, Košice. He was a member of the CDF collaboration at the Fermi National Laboratory and worked as a technical student at the European Organization for Nuclear Research (CERN). He is the author and co-author of several scientific papers and participates in the EU IST projects K-Wf Grid, EGEE-2, Degree, as well as in several national projects. He is experienced in the Semantic Web technologies and languages; development of large scale systems; development of knowledge based systems and semantic web services.



Ladislav HLUCHÝ is the director of the Institute of Informatics of the Slovak Academy of Sciences and also the head of the Department of Parallel and Distributed Computing at the institute. He received M. Sc. and Ph. D. degrees, both in computer science. He is R&D Project Manager, Work-package Leader in a number of 4FP, 5FP and 6FP projects, as well as in Slovak R&D projects (VEGA, APVT, SPVV). He is a member of IEEE, ERCIM, SRCIM, and EuroMicro consortiums, the editor-in-chief of the journal Computing and Informatics. He is also (co-)author of scientific books and numerous scientific papers, contributions and invited lectures at international scientific conferences and workshops. He also gives lectures at Slovak University of Technology and is a supervisor and consultant for Ph. D., master and bachelor studies.