A Semantic Framework for Grid-based Service Registries

Marian Babik, Ladislav Hluchý

Department of Parallel and Distributed Computing
Institute of Informatics, Slovak Academy of Sciences
E-mail: Marian.Babik@saske.sk

Abstract

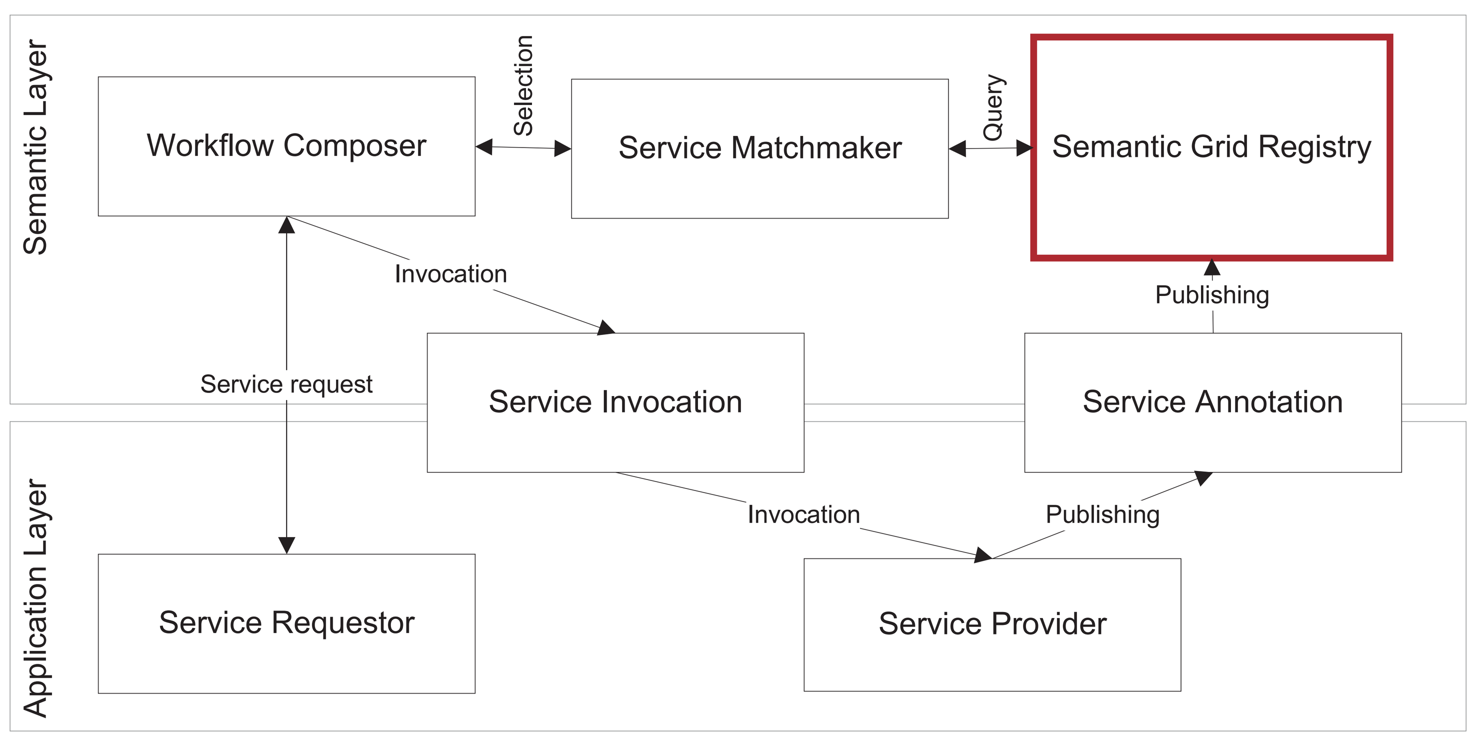
The Semantic Grid is a recent initiative to expose semantically rich information associated with Grid resources to build more intelligent Grid services [1]. Recently, several projects have embraced this vision and there are several successful applications that combine the strengths of the Grid and of semantic technologies. However, Semantic Grid still lacks a technology, which would provide the needed scalability and integration with existing infrastructure.
In this paper we present our on-going work on a semantic grid 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.

Motivation

Creation and management of the distributed and heterogenous resources are the key challenges in several scientific applications such as bioinformatics, chemistry and environmental sciences. 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. Such environments often require support for automated discovery, matchmaking, composition and executions of the grid and web services. Such functionality is envisioned in the context of the so called Semantic Web Services (SWS). Generally, all the SWS-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, accessing the knowledge modeled in the ontologies, is not limited to syntactical matching, but it can also exploit the existing semantic matching.

Although there are various different approaches to the service matchmaking and composition, there are only few typical query types, which the underlying semantic repository has to support



The current Semantic Web infrastructure offers a range of existing ontological repositories, which can provide the support for ontological querying and reasoning. However, there are several shortcomings in the current implementations:

1. The well-known repositories such as Sesame, Kowari and Jena2 are mostly based on the RDF/RDFS and thus provide only models with limited expressivity and no or very restricted support for OWL ontologies.
2. Conjunctive queries over ontologies with large number of instances is only available on certain system.
3. There is a limited support for performing ontological queries over relational databases. This is a major obstacle in performing an integration of the semantic repositories with existing relational-based systems (such as UDDI, RGMA, etc.).

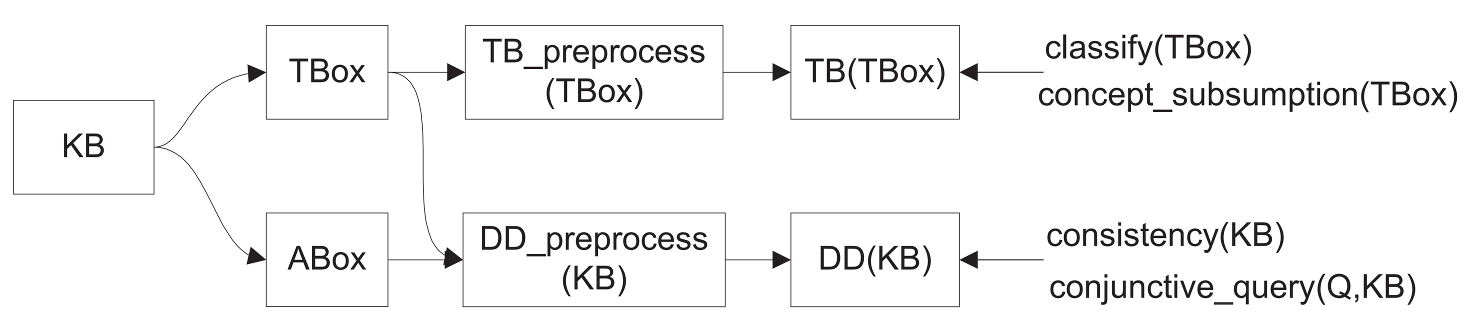
In summary, our goal is to provide a scalable ontological repository for services and data, which would support both subsumption and conjunctive queries as well as coupling with legacy metadata stored in relational databases.

Approach

The Semantic Web proposes a standard for ontological descriptions in the form of a Web Ontology Language(OWL). There are currently two algorithms for reasoning with OWL ontologies, namely, tableau decision procedure [2] and reasoning in the framework of resolution [3]. In the following we will denote the tableau procedure as TB and the resolution based procedure with DD.
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:

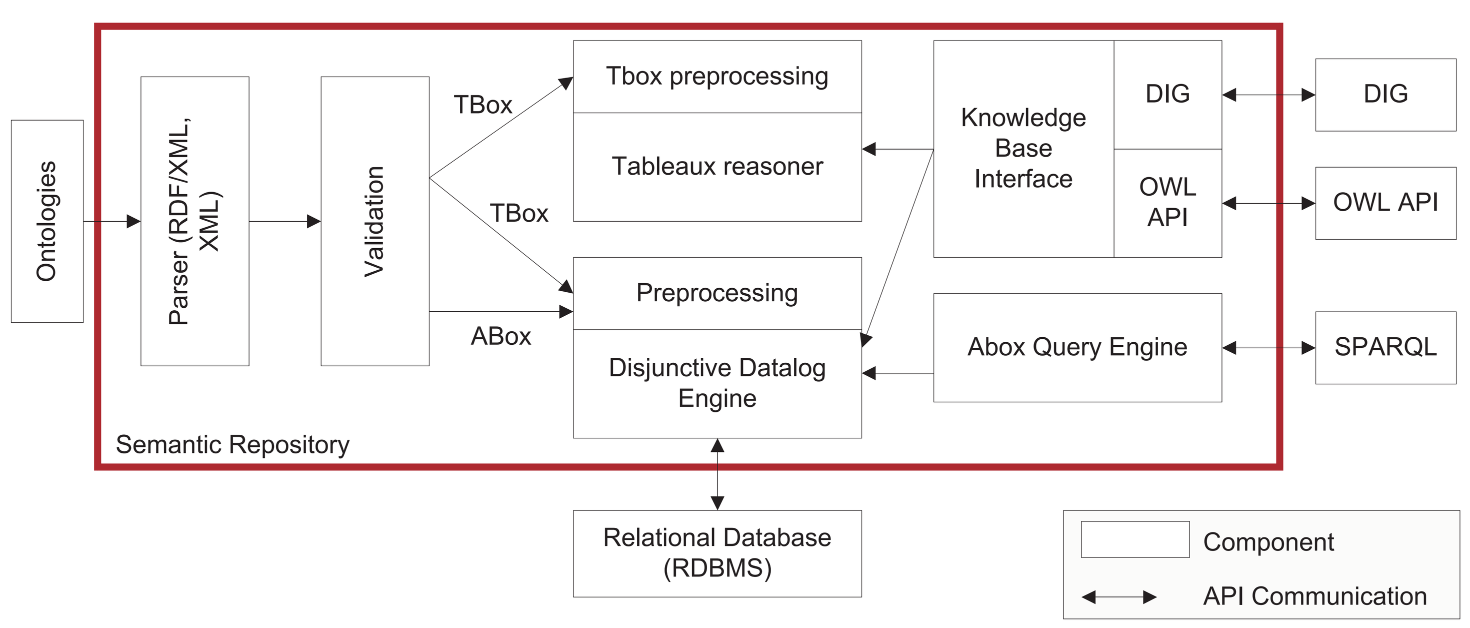
- Our method combines the existing description logics reasoners for computing taxonomies (TBoxes), i.e. TB, with rule-based reasoners for the reasoning with large number of instances (ABoxes), i.e. DD.
- Based on the proposed combination we can re-use the existing optimizations (i.e. classification and 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 grid registries.



In our approach we initially split the KB into a set of TBox and ABox assertions and feed these assertions to different reasoners as follows. The overall Tbox is loaded into a tableau-based reasoner, i.e. TB(TBox). This allows to ask complex schema queries and perform classification of the TBox without considering the Abox, which for the tableau methods is quite costly. Subsequently, the overall KB is then loaded to the disjunctive datalog engine, i.e. DD(KB). This enables the possibility to check the consistency of the overall KB, but to leave the TBox specific tasks to the tableau reasoners. The subsequent classification and taxonomy queries can be forwarded to the TB(TBox), while the conjunctive queries and consistency of the KB can be handled by the DD(KB). Since the decomposition of the KB to the TBox and ABox has to be performed anyway, there is no significant performance overhead during the initialization of the reasoners.

Implementation



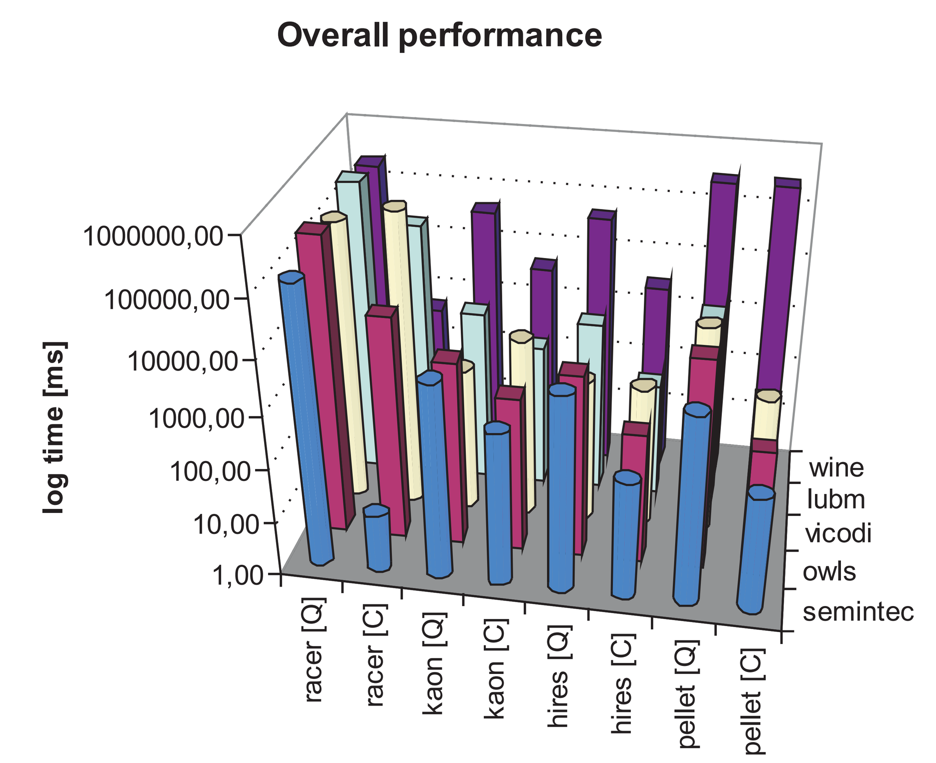
The overall architecture of the system is based on extending the existing tableau reasoner with the optimizations for the conjunctive query answering and database backend. Figure above shows the main components of the system. The core of the system is composed of two reasoners, tableau reasoner and disjunctive 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 and its aim is to check the consistency of the knowledge base KB and to perform the conjunctive queries over ABox [3]. The system supports OWL API, SPARQL queries and DIG interface.

Performance Evaluation

Evaluation of our approach is based on the comparison of the performance of our system with that of existing description logics reasoners. We compared our system (denoted Hires) with well-known description logic reasoners KAON2, Racer and Pellet. The tests were run on the ontologies summarized in Tab.1. For all the ontologies we have measured the time (in ms) to answer a conjunctive query (query time Q) and the time to compute the classification of the ontology (classification time C). All tests were performed on the laptop computer (T60) with 1.8Ghz memory and 1 GB of RAM, running Linux kernel 2.6.20. For Java-based reasoners (Pellet, KAON2) we have used Java runtime 1.5.0 Update 6 with virtual memoryrestricted 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 either run out of memory or out of time are denoted with time 300000.

KB	concepts	roles	concept instances	role instances
semintec	168	6	89705	236240
vicodi	193	10	67768	146844
lubm	42	9	83200	236514
wine	188	9	79287	78966
owls	92	9	120000	0
dolce	272	522	0	0
galen	3936	287	0	0

Tab. 1 Statistics of the benchmark ontologies



	racer [Q]	racer [C]	kaon [Q]	kaon [C]	hires [Q]	hires [C]	pellet [Q]	pellet [C]
semintec	138930,40	12,00	4345,70	745,80	4406,40	159,40	3366,30	148,60
owls	300000,00	14122,00	2635,00	757,00	2561,00	277,60	8415,40	225,00
vicodi	162838,70	300000,00	494,70	2238,60	458,20	449,60	7591,20	462,60
lubm	300000,00	60011,80	1680,47	467,80	1626,40	141,20	3533,20	108,00
wine	233420,80	495,40	45064,20	4600,00	48653,80	2935,60	300000,00	300000,00

semintec owls vicodi lubm wine

Conclusion

We have described design and development of the scalable semantic grid registry [4]. 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 int.eu.grid. In the future we would like to address caching and materialization aspects of the semantic storage.

Acknowledgments: The research reported in this paper has been partially financed by the EU within the project EU 6FP RI-031857 int.eu.grid and Slovak national projects, NAZOU SPVV 1025/2004, SEMCO-WS APVV-0391-06, APVV RPEU-0024-06, APVV RPEU-0029-06, VEGA 2/6103/6, VEGA 2/7098/27.

References

[1] Goble, C., De Roure, D., The Semantic Grid: Myth Busting and Bridge Building, in Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004), Valencia, Spain, 2004
[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press 2003, ISBN 0-521-78176-0
[3] Y. Kazakov, B. Motik. A Resolution-Based Decision Procedure for SHOIQ. In Ulrich Furbach, John Harrison, and Natarajan Shankar, editors, Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR 2006), volume 4130 of LNAI, pages 662-667, Seattle, WA, USA, August 17-20 2006. Springer.
[4] Babik, M.: Optimizing classification and instance retrieval in description logic reasoning. Technical report, Institute of Informatics, SAS, 2007
http://www.tuke.sk/fei-cit/babik/report.pdf

Cracow Grid Workshop, October, 2007, Cracow